

UNIVERSITATEA BUCUREŞTI
Facultatea de Matematică și Informatică
Programul de studii: *Informatică*



**UNIVERSITATEA DIN
BUCUREŞTI**
— VIRTUTE ET SAPIENTIA —

Disciplina: Baze de date

GESTIUNEA CINEMATOGRAFELOR

COORDONATOR ȘTIINȚIFIC:
Lect.dr. Letiția-Ana MARIN
Asist.drd. Natalia-Gabriela OZUNU

Student:
Ştefania RÎNCU

Grupa 132

BUCUREŞTI
2022

Cuprins

1.	Descrierea modelului real, a utilității acestuia și a regulilor de funcționare	3
2.	Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.....	4
3.	Descrierea entităților, incluzând precizarea cheii primare	5
4.	Descrierea relațiilor.....	6
5.	Descrierea atributelor.....	9
6.	Diagrama Entitate-Relație.....	15
7.	Diagrama Conceptuală	16
8.	Schemele relaționale	17
9.	Realizarea normalizării până la forma normală 3 (FN1-FN3).....	18
10.	Crearea secvențelor	20
11.	Crearea tabelelor, inserarea datelor.....	26
12.	Cereri SQL.....	65
13.	Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.....	71
16.	Implementarea operației de outer join și cererile ce utilizează operația division	73
17.	Optimizări și arbori algebrici	75
18.	Normalizări BCNF, FN4, FN5 + denormalizare	78

1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare

Bazele de date sunt foarte utile în gestionarea centralizată și în timp real a informațiilor din diverse domenii de activitate. În cadrul acestei lucrări se va prezenta proiectarea unei baze de date care să se ocupe cu gestiunea cinematografelor dintr-o țară. În acest domeniu o bază de date ar fi utilă în gestionarea informațiilor pentru organizarea procesului de administrare a locațiilor unde se vor viziona filmele incluând și personalul necesar pentru deservirea acestora, dar și din punctul de vedere al materialelor ce sunt prezentate în cadrul cinematografelor inclusiv filme și personalul care s-a implicat în producerea acestoara (regizori, actori).

Pentru proiectarea bazei de date am impus o serie de reguli necesare pentru buna funcționare a acesteia. Regulile de funcționare ale acestui model sunt următoarele:

1. În diagramă sunt incluse doar cinematografele care rulează filme
2. Cinematografele rulează filme din întreaga lume
3. În acest model se presupune că cel puțin o persoană trebuie să lucreze într-un cinematograf ca acesta să existe (spre exemplu: managerul)
4. Angajații pot fi: casieri (se ocupă cu vânzarea de bilete, rezervări), plasatori (se ocupă cu direcționarea corectă a clientilor în cinema, verificarea biletelor de acces, implementarea politicii referitoare la restricțiile de vârstă), barman cafenea sau barman (se ocupă cu aranjarea, prepararea și servirea produselor de la bar), agent de curățare (se ocupă cu menținerea curățeniei), ingineri sunet, ingineri tehnic, manager
5. Joburile sunt: casier, plasator, barman cafenea, barman, agent de curățare, inginer sunet, inginer tehnic, manager
6. Actorii sunt persoanele care joacă într-un film, fiind gestionăți actorii mai cunoscuți, despre care știm că vor stârni interesul publicului
7. Pentru actori și regizori vrem să știm țara natală
8. Ca și premii pentru actori și regizori se vor număra doar premiile importante
9. Un cinematograf se poate afla în cadrul unui complex comercial
10. În diagramă sunt inclusi doar angajații care lucrează într-un cinematograf
11. În diagramă sunt incluse și filmele pentru care nu avem detalii despre difuzare sau rulare
12. În diagramă sunt incluse și locații în care nu există cinematografe

2. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului

Deoarece această diagramă nu oglindește perfect realitatea am introdus o serie de constrângeri pe care le-am aplicat asupra modelului, astfel încât datele pe care le voi introduce și relațiile descrie să fie coerente.

Restricțiile și regulile impuse asupra modelului sunt următoarele:

1. Un cinematograf se poate afla într-o singură locație
2. Un angajat poate lucra pentru un singur cinematograf la un moment dat de timp
3. Un cinematograf poate avea minim 1 angajat (atunci când abia se deschide), acesta fiind managerul cinematografului
4. În diagramă sunt incluse doar cinematografele care rulează filme
5. Un job aparține de un singur cinematograf la un moment dat de timp
6. Un cinematograf poate avea minim un job, deoarece întotdeauna va exista managerul
7. La o locație nu putem avea mai multe cinematografe, dar putem să nu avem niciunul
8. Un film poate fi îndrumat de mai mulți regizori
9. Un regizor poate îndruma minim 1 film, deoarece altfel nu îl putem clasifica drept regizor recunoscut
10. Un film poate exista fără niciun actor, dacă este un film animat
11. Un actor trebuie să apară în minim un film pentru a fi recunoscut ca actor
12. O sală poate avea minim 1 loc, altfel nu au unde să se așeze spectatorii și nu putem vorbi despre o sală de difuzare a filmelor
13. Pe bilet apare codul difuzării căreia îi este asignat,
14. Atât actorii, cât și regizorii pot avea o singură țară natală
15. O difuzare poate avea minim 0 bilete, dacă nicio persoană nu vine să o vizioneze

3. Descrierea entităților, incluzând precizarea cheii primare

Baza de date proiectată în cadrul acestei lucrări conține entități ce desemnează componente care ajută la favorizarea bunei funcționări a unui cinematograf (spre exemplu sălile din acesta sau angajații care muncesc acolo).

Entitățile folosite în realizarea acestui proiect sunt:

1. **CINEMATOGRAF**(entitate independentă) – loc cu săli de spectacole destinate proiecției filmelor cinematografice în fața publicului spectator, cheia primară este „**cod_cinema**”
2. **SALA** (entitate dependentă) –încăperea în care sunt proiectate filmele în fața spectatorilor, nu poate exista fără să aparțină de un cinematograf, cheia primară este compusă din „**cod_cinema**” (cheie străină, ce aparține entității CINEMATOGRAF) și „**nr_sala**”
3. **ANGAJAT** (entitate independentă) – persoana care lucrează în cadrul cinematografului și care servește la buna funcționare a acestuia, cheia primară este „**cod_angajat**”
4. **JOB** (entitate independentă) –meseria pe care o poate avea un angajat, cheia primară este „**cod_job**”
5. **LOCATIE** (entitate independentă) – adesa completă unde se află un cinematograf, descriindu-se poziționarea geografică, cheia primară este „**id_locatie**”
6. **FILM** (entitate independentă) –înregistrare ce cuprinde momente succesive constând în imagini statice sau în mișcare, precum și sunete, cu scopul de a fi mai apoi proiectate pe un ecran, cheia primară este „**cod_film**”
7. **BILET** (entitate independentă) -bucată mică de hârtie sau de carton imprimat care dă dreptul la intrarea, ocuparea unui loc într-o sală și vizionarea unui film, cheia primară este „**cod_bilet**”
8. **DIFUZARE** (entitate dependentă) – acțiunea de a proiecta un film , într-o sală dintr-un cinematograf, cheia primară este „**cod_difuzare**”
9. **ACTOR** (entitate independentă) - persoană care interpretează diferite roluri în filme, cheia primară este „**id_actor**”
10. **REGIZOR** (entitate independentă) - specialist cu calificare profesională care se ocupă cu regia filmelor, cheia primară este „**id_regizor**”

4. Descrierea relațiilor

Descrierea relațiilor care au fost folosite în cadrul acestui proiect respectă regulile modelului și constrângerile impuse pentru o bună funcționare a sistemului de gestiune a bazei de date.

Detalierea acestor relații, incluzând precizarea cardinalității acestora:

1. **detine** (CINEMATOGRAF, SALA) – relație ce indică faptul că o sală aparține de un cinematograf, fiind o relație de dependență, deoarece o sală (după cum am definit-o, ca fiind special creată pentru proiecțarea filmelor) nu poate exista fără un cinematograf. Cardinalitățile sunt deduse din întrebările: „Câte săli poate detine un cinematograf? **M**. Câte săli trebuie să detină un cinematograf? Cel puțin **1**, deoarece altfel nu ar avea unde să difuzeze filme. În câte cinematografe se poate afla o sală? În câte cinematografe trebuie să se afle o sală? În **1** cinematograf (pentru ambele întrebări).”
=>**Cardinalitate:M(1)** (Cinematograf –Sală), **1** (Sală – Cinematograf)
2. **ruleaza** (CINEMATOGRAF, FILM) – relație ce indică faptul că un film poate fi rulat într-un cinematograf. Cardinalitățile sunt deduse din întrebările: „Câte filme pot rula într-un cinematograf? **M**. Câte filme trebuie să ruleze într-un cinematograf? Cel puțin **1**, însăcum în această diagramă sunt incluse doar cinematografele care rulează filme. În câte cinematografe poate rula un film? **M**. În câte cinematografe trebuie să ruleze un film? **0**, dacă încă nu avem detalii despre rulare.”
=>**Cardinalitate:M(1)** (Cinematograf –Film), **M(0)** (Film- Cinematograf)
3. **are_loc** (SALA, DIFUZARE) – relație ce indică faptul că fiecare difuzare din cinematograf are loc într-o sală. Cardinalitățile sunt deduse din întrebările: „Câte difuzări pot avea loc într-o sală? **M**. Câte difuzări trebuie să aibă loc într-o sală? **0**, deoarece există posibilitatea să nu difuzeze nimic în acel moment. În câte săli poate avea loc o difuzare? În câte săli trebuie să aibă loc o difuzare? **1** (pentru ambele întrebări) deoarece pentru a vorbi despre o difuzare, aceasta trebuie să aibă loc într-o sală.”
=>**Cardinalitate:M(0)** (Sală - Difuzare), **1** (Difuzare - Sală)
4. **se_proiecteaza**(FILM, DIFUZARE) – relație ce indică că fiecare film se proiectează în cadrul unei difuzări. Cardinalitățile sunt deduse din întrebările: „La câte difuzări se poate proiecta un film? **M**. La câte difuzări trebuie să se proiecteze un film? Cel puțin **1**, deoarece în această diagramă sunt incluse filmele ce au fost difuzate cel puțin o dată. Câte filme se pot proiecta la o difuzare? Câte filme trebuie să se proiecteze la o difuzare? **1** (pentru ambele întrebări) deoarece pentru a vorbi despre o difuzare, aceasta trebuie să prezinte un film.”
=>**Cardinalitate:M(0)** (Film - Difuzare), **1**(Difuzare - Film)
5. **se_intra** (BILET, DIFUZARE) – relație ce indică faptul că pe baza unui bilet se intră la o difuzare de film. Cardinalitățile sunt deduse din întrebările: „La câte difuzări se

poate intra pe baza unui bilet? La câte difuzări trebuie să se intre pe baza unui bilet? **1** (pentru ambele întrebări). Cu câte bilete se poate intra la o difuzare? **M** (cumpărate de mai multe persoane pentru aceeași difuzare). Cu câte bilete trebuie să se intre la difuzare? **0**, deoarece există posibilitatea să nu cumpere nimeni bilet.”

=>**Cardinalitățile:1** (Bilet - Difuzare), **M(0)** (Difuzare - Bilet)

6. **indruma** (REGIZOR, FILM) – relație ce ne indică faptul că filmele sunt îndrumate de către regizori. Cardinalitățile sunt deduse din întrebările: „Câte filme pot fi îndrumate de un regizor? **M**. Câte filme trebuie să fie îndrumate de un regizor? Cel puțin **1**, întrucât am stabilit că un regizor trebuie să aibă măcar un film regizat pentru a fi recunoscut. Câți regizori pot să îndrume un film? **M**. Câți regizori trebuie să îndrume un film? **0**, deoarece putem vorbi despre un film animat.”
=>**Cardinalitățile:M(0)** (Film - Regizor), **M(1)** (Regizor - Film)
7. **joaca** (ACTOR, FILM) – relație ce indică faptul că actorii trebuie să aibă filme în care să joace pentru a fi recunoscuți, iar filmele au nevoie de actori care să dea contur acțiunii. Cardinalitățile sunt deduse din întrebările: „În câte filme poate să joace un actor? **M**. În câte filme trebuie să joace un actor? Cel puțin**1**, deoarece am stabilit că pentru a fi recunoscut un actor trebuie să joace în cel puțin un film. Câți actori pot juca într-un film? **M**. Câți actori trebuie să joace într-un film? **0**, deoarece putem vorbi despre un film animat.”
=>**Cardinalitățile:M(1)** (Actor – Film), **M(0)** (Film – Actor)
8. **lucreaza** (CINEMATOGRAF, ANGAJAT, JOB) – relație de tip 3 ce indică fapul că există o legătură între angajatul care are un anumit job și lucrează în cadrul unui cinematograf. Cardinalitățile sunt deduse din întrebările: „Câți angajați poate lucra într-un cinematograf? Câți angajați pot lucra pe un anumit job? **M**. Câți angajați trebuie să lucreze într-un cinematograf? Câți angajați trebuie să lucreze pe un anumit job? Minim **1** angajat, întrucât există situația în care abia s-a deschis cinematograful și există doar managerul și ținem cont de fapul că fără un angajat nu putem vorbi despre un job. Câte joburi poate avea un cinematograf? Câte joburi poate lucra un angajat? **M**. Câte joburi trebuie să aibă un cinematograf? Câte joburi trebuie să lucreze un angajat? Minim **1** job, întrucât întotdeauna va exista jobul de manager, iar pentru ca jobul să fie declarat atunci trebuie să lucreze cel puțin un angajat. În câte cinematografe poate lucra un angajat? În câte cinematografe se poate lucra un anumit job? În câte cinematografe trebuie să lucreze un angajat? În câte cinematografe trebuie să se lucreze un anumit job? **1** (pentru toate cele 4 întrebări), deoarece un angajat nu poate lucra la mai multe cinematografe simultan, iar pentru a fi angajat trebuie să lucreze la un cinematograf.”
=>**Cardinalitățile:M(1)** (Cinematograf - Angajat, Job - Angajat), **M(1)** (Angajat – Job, Cinematograf - Job), **1**(Angajat – Cinematograf, Cinematograf - Job)
9. **se_afla**(CINEMATOGRAF, LOCATIE) – relație ce indică poziționarea unui cinematograf, arătându-ne adresa acestuia. Cardinalitățile sunt deduse din întrebările: „Câte cinematografe se poate afla într-o locație? **1**. Câte cinematografe trebuie să se afle la o locație? **0**, întrucât într-un punct anume se poate să nu fie niciun cinematograf construit. În câte locații se poate afla un cinematograf? În câte locații trebuie să se

afle un cinematograf? **1** locație (pentru ambele întrebări), deoarece un cinematograf nu se poate afla simultan în mai multe locații, iar pentru a exista atunci trebuie să se regăsească undeva.”

=>**Cardinalitățile:1(0)** (Locație - Cinematograf), **1** (Cinematograf – Locație)

5. Descrierea atributelor

Descrierea atributelor s-a făcut ținând cont și de tipul de date, eventualele constrângeri, valori implicate, valori posibile ale acestora.

Atributele ce se regăsesc în cadrul acestui proiect sunt:

1. CINEMATOGRAF

- a. **cod_cinema** - număr de maxim 4 cifre (NUMBER(4)), care reprezintă codul cinematografului, fiind unic pentru fiecare cinematograf în parte (exemplu: 132116). Aceasta reprezintă cheia primară a entității CINEMATOGRAF
- b. **nume_cinema** – sir de maxim 25 de caractere (litere mari, mici, „_”) (VARCHAR(25)), care precizează numele cinematografului, aşa cum se regăsește în acte (exemplu: Happy_Cinema). Aceasta este diferit de „null”, deoarece un cinematograf care există obligatoriu are un nume
- c. **id_locatie** – număr de maxim 4 cifre (NUMBER (4)), care precizează id-ul locației unde se regăsește cinematograful (exemplu: 113). Aceasta este diferit de „null”, deoarece dacă un cinematograf există atunci acesta se regăsește într-o locație. Acest id are corespondent în tabela LOCATIE („id_locatie”)
- d. **complex** - sir de maxim 30 de caractere (VARCHAR2(30)), care dă denumirea complexului comercial în care cinematograful se regăsește (litere mari, mici, „_”) (exemplu: Afi_Palace_Cotroceni)
- e. **telefon** - sir de maxim 20 de caractere(VARCHAR2(20)), precizează numărul de telefon la care poate fi contactat cinematograful (exemplu: 0741.321.133)
- f. **email** – sir de maxim 25 de caractere(litere mari, mici, „_”, „@”, „.”, cifre)(CHAR(25)), precizează emailul la care poate fi contactat cinematograful (exemplu: happy.cinema@gmail.com). Aceasta trebuie să fie diferit de valoarea „null” și este unic
- g. **data_desch** - este o dată calendaristică (DATE) care arată data în care s-a deschis cinematograful respectiv (exemplu: 3 mai 2022). Aceasta este diferită de „null” și are valoarea „sysdate” by default

2. SALA

- a. **cod_cinema** - număr de maxim 4 cifre (NUMBER(4)), care reprezintă codul cinematografului în care se află sala respectivă, codul fiind unic. Aceasta reprezintă ocheie primară externă a entității SALA și are corespondent în tabela CINEMATOGRAF, având o valoare diferită de „null”
- b. **nr_sala** - număr de maxim 2 cifre (NUMBER(2)), mai mic decât 40 și mai mare ca 0, care reprezintă numărul sălii. Aceasta reprezintă cheia primară a entității SALA, împreună cu atributul „**cod_cinema**”

c. **tip** –șir de maxim 10 caractere (VARCHAR(10)) care definește tipul unei săli de cinema (exemplu: VIP)

3. FILM

- a. **cod_film** - număr de maxim 4 cifre (NUMBER(4)), care reprezintă codul filmului, acesta fiind unic pentru fiecare film în parte (de exemplu: 1652). Acesta reprezintă cheia primară a entității FILM
- b. **nume_film** - șir de maxim 25 de caractere (litere mari, mici, „_”, cifre) (VARCHAR (25)), care precizează numele complet al filmului, aşa cum se regăseşte pe afişul filmului (exemplu: Uncharted). Acesta este diferit de „null”, deoarece un film care există obligatoriu are un nume și este unic (în asociere cu anul apariției)
- c. **gen** - șir de maxim 20 de caractere (VARCHAR (20)), care precizează genul filmului (litere mari, mici, „ , ”) (exemplu: Actiune). Acesta este diferit de „null”, deoarece orice film aparține unei tipologii
- d. **durata** – număr de maxim 3 cifre (NUMBER(3)), care precizează durata filmului (în minute) (exemplu: 93). Acesta are valoarea implicită 0 minute
- e. **rating** - număr cu virgulă (de tipul „cifră.cifră”), float (NUMBER(2, 1)), care precizează aprecierea publicului (exemplu: 7.5), acesta are o valoare din intervalul 0 - 10
- f. **an_aparitie** - număr de 4 cifre, int (NUMBER(4)), care reprezintă anul în care a apărut filmul respectiv (exemplu: 2020). Acesta este diferit de „null” și mai mare decât 0, deoarece pentru ca un film să apară la cinematograf atunci trebuie să aibă un an în care a fost difuzat prima dată și unic

4. DIFUZARE

- a. **cod_difuzare** - număr de maxim 4 cifre (NUMBER(4)), care reprezintă codul difuzării, acesta fiind unic pentru fiecare difuzare în parte în parte (de exemplu: 1655). Acesta reprezintă cheia primară a entității DIFUZARE
- b. **cod_film** - număr de maxim 4 cifre (NUMBER(4)), care reprezintă codul filmului (de exemplu: 1652). Acesta este diferit de „null” și referențiază cheia primară a entității FILM
- c. **nr_sala** - număr de maxim 2 cifre (NUMBER(2)), mai mic sau egal decât 40 și mai mare ca 0, care reprezintă numărul sălii. Acesta este diferit de „null”
- d. **cod_cinema** – număr de maxim 4 cifre (NUMBER(4)), care reprezintă codul cinematografului în care se află difuzarea respectivă. Acesta are o valoare diferită de „null” și împreună cu atributul „nr_sala” referențiază cheia primară a entității SALA
- e. **data_inc** - este o dată calendaristică (DATE) care arată data în care are loc difuzarea respectivă (exemplu: 3 mai 2022). Aceasta este diferită de „null” și are valoarea „sysdate” by default
- f. **ora_inc** - număr de maxim 2 cifre (NUMBER(2)), care reprezintă ora la care începe difuzarea respectivă (exemplu: 7). Acesta este diferit de „null”, mai mare ca 0 și mai mic decât 24

5. BILET

- a. **cod_bilet** – număr de maxim 4 cifre (NUMBER(4)), care reprezintă codul biletului vândut, acest cod fiind unic. Aceasta reprezintă cheia primară a entității BILET și nu poate avea valoarea „null”, deoarece atunci când este vândut un bilet primește un cod
- b. **pret** - număr cu virgulă (de tipul „cifre.cifre”), float (NUMBER(8, 2)), care precizează prețul biletului achiziționat, preț care diferă de la film la film (exemplu: 18.50). Aceasta este mai mare sau egal decât 0.0, având valoare implicită 0.0
- c. **nr_loc** – număr de maxim 3 cifre (NUMBER(3)), mai mic sau egal decât 300 și mai mare ca 0, precizează numărul locului pe care îl va avea asignat spectatorul care se află în posesia aceluia bilet. Aceasta nu poate avea valoarea „null”, deoarece o dată cumpărat, orice bilet are atribuit un loc pentru cumpărător
- d. **nr_rand** – număr de maxim 2 cifre (NUMBER(2)), mai mic sau egal decât 20 și mai mare ca 0, precizează numărul rândului pe care se află locul pe care îl va avea asignat spectatorul care se află în posesia aceluia bilet. Aceasta nu poate avea valoarea „null”
- e. **data_cump** - este o dată calendaristică (DATE) care arată data în care a fost achiziționat biletul respectiv (exemplu: 3 mai 2022). Aceasta este diferită de „null” și are valoarea „sysdate” by default
- f. **cod_diffuzare** - număr de maxim 4 cifre (NUMBER(4)), care reprezintă codul difuzării căreia îi este asignat acest bilet. Aceasta are valoarea diferită de „null” și referențiază cheia primară a entității DIFUZARE („cod_diffuzare”)

6. ANGAJAT

- a. **cod_angajat** - număr de maxim 4 cifre (NUMBER(4)), care reprezintă codul angajatului, acest cod fiind unic (exemplu: 420). Aceasta reprezintă cheia primară a entității ANGAJAT
- b. **nume** - sir de maxim 25 de caractere (VARCHAR (25)), care precizează numele de familie al angajatului, aşa cum se regăseşte în cartea de identitate a acestuia (litere mari, mici, „_”) (exemplu: Popescu). Aceasta este unic și diferit de „null”, deoarece un angajat trebuie să apară în baza de date măcar cu numele de familie
- c. **prenume** - sir de maxim 25 de caractere (litere mari, mici, „_”)(VARCHAR (25)), care precizează prenumele angajatului, aşa cum se regăseşte în cartea de identitate a acestuia.
- d. **email** – sir de maxim 25 de caractere (CHAR(25)) (litere mari, mici, „_”, „@”, „.”, cifre), precizează emailul la care poate fi contactat angajatul (exemplu: popescu@gmail.com). Aceasta trebuie să fie diferit de valoarea „null” și este unic

- e. **telefon** - sir de maxim 20 de caractere (VARCHAR2(20)), precizeaza numarul de telefon la care poate fi contactat angajatul, dar acesta nu este obligat sa ofere aceasta informatie (exemplu: 0723.999.234)
- f. **nr_ore** – numar de maxim 2 cifre (NUMBER(2)), care reprezinta numarul de ore pe care le lucreaza angajatul respectiv intr-o zi (norma standard, nu se tine cont de orele suplimentare) (exemplu: 7). Aceasta este diferit de „null”, mai mare ca 0 si mai mic decat 24
- g. **id_superior** - de maxim 4 cifre (NUMBER(4)), care reprezinta codul superiorului fiecarui angajat (exemplu: 420). Aceasta poate primi valori din multimea „cod_angajat” din tabela ANGAJAT, deoarece un superior trebuie sa fie angajat la randul sau

7. JOB

- a. **cod_job** - sir de maxim 6 caractere (VARCHAR(6)), care reprezinta codul jobului, format din initialele reprezentative ale acestui job, acest cod fiind unic (exemplu: CAS (= casier), contraexemplu: 117). Aceasta reprezinta cheia primara a entitatii JOB
- b. **nume_job** - sir de maxim 20 de caractere (litere mari, mici, „_”) (VARCHAR(20), care reprezinta numele jobului, asa cum apare acesta in fisiera de post (exemplu: casier). Aceasta nu poate avea valoarea „null” si este unic
- c. **salariu_min** - numar cu virgula (de tipul „cifre.cifre”) (NUMBER(8,2)), care reprezinta salariul minim asignat jobului respectiv (exemplu: 1000.9). Aceasta este diferit de „null” si mai mare decat 0, deoarece un salariu nu poate fi negativ
- d. **salariu_max** - numar cu virgula (de tipul „cifre.cifre”) (NUMBER(8,2)), care reprezinta salariul maxim asignat unui job (exemplu: 6000.9). Aceasta este diferit de „null” si mai mare decat 0, deoarece un salariu nu poate fi negativ

8. LOCATIE

- a. **id_locatie** – numar de maxim 4 cifre (NUMBER(4)), care reprezinta id-ul locatiei, acest id fiind unic (exemplu: 177). Aceasta reprezinta cheia primara a entitatii LOCATIE
- b. **tara** - sir de maxim 3 caractere (VARCHAR(3)), format din initialele tarii respective, care reprezinta id-ul tarii, acest id fiind unic (exemplu: RO). Aceasta nu poate avea valoarea „null”
- c. **oras** – sir de maxim 20 de caractere (litere mari, mici, „_”) (VARCHAR(20), care reprezinta numele orasului unde se regaseste o anumita locatie (exemplu: Bucuresti). Aceasta nu poate avea valoarea „null”
- d. **localitate** – sir de maxim 20 de caractere (litere mari, mici, „_”) (VARCHAR(20), care reprezinta numele localitatii unde se regaseste o anumita locatie (exemplu: Bilcesti)

- e. **strada** – sir de maxim 20 de caractere (litere mari, mici, „_”)(VARCHAR(20)), care reprezinta numele strazii pe care se regaseste o anumita locatie (exemplu: Arefu). Aceasta nu poate avea valoarea „null”
- f. **nr_strada** – numar de maxim 3 cifre (NUMBER(3)), care indica numarul strazii pe care se regaseste o anumita locatie (exemplu: 59)

9. ACTOR

- a. **id_actor** – numar de maxim 4 cifre (NUMBER(4)), care reprezinta id-ul unui actor, acest id fiind unic (exemplu: 777123). Aceasta reprezinta cheia primara a entitatii ACTOR si nu poate avea valoarea „null”
- b. **nume** – sir de maxim 25 de caractere (litere mari, mici, „_”)(VARCHAR (25)), care precizeaza numele de familie al actorului sau numele acestuia de scenă, după cel care este mai recunoscut (exemplu: Depp sau Coluche). Aceasta este diferit de „null”, deoarece trebuie sa existe un nume după care să poată fi identificat, fiind unic
- c. **prenume** – sir de maxim 25 de caractere (litere mari, mici, „_”)(VARCHAR (25)), care precizeaza prenumele actorului, fiind optională precizarea acestuia
- d. **tara** – sir de maxim de 3 caractere (VARCHAR(3)), format din literele reprezentative ce alcătuiesc numele țării respective, reprezentând id-ul țării natale a actorului respectiv (exemplu: RO). Aceasta este unic și diferit de „null”, deoarece orice persoană trebuie să aibă o țară natală
- e. **nr_premii** – numar de maxim 3 cifre (NUMBER(3)), care reprezintă numarul de premii importante (Oscar, Grammy etc) primite de actorul respectiv până în prezent (exemplu: 3). Are valoare implicită 0 și este mai mare sau egal decât 0
- f. **an_debut** – numar de 4 cifre (NUMBER(4)), care reprezintă anul în care și-a făcut debutul actorul respectiv (exemplu: 1999). Aceasta este diferit de „null”, deoarece pentru ca un actor să fie recunoscut trebuie să debuteze mai întâi, de asemenea, este mai mare ca 0
- g. **salariu** - numar cu virgulă (de tipul „cifre.cifre”) (NUMBER(8, 2)), care reprezintă salariul unui actor (exemplu: 56212.21). Aceasta este diferit de „null” și mai mare decât 0, deoarece un salariu nu poate fi negativ

10. REGIZOR

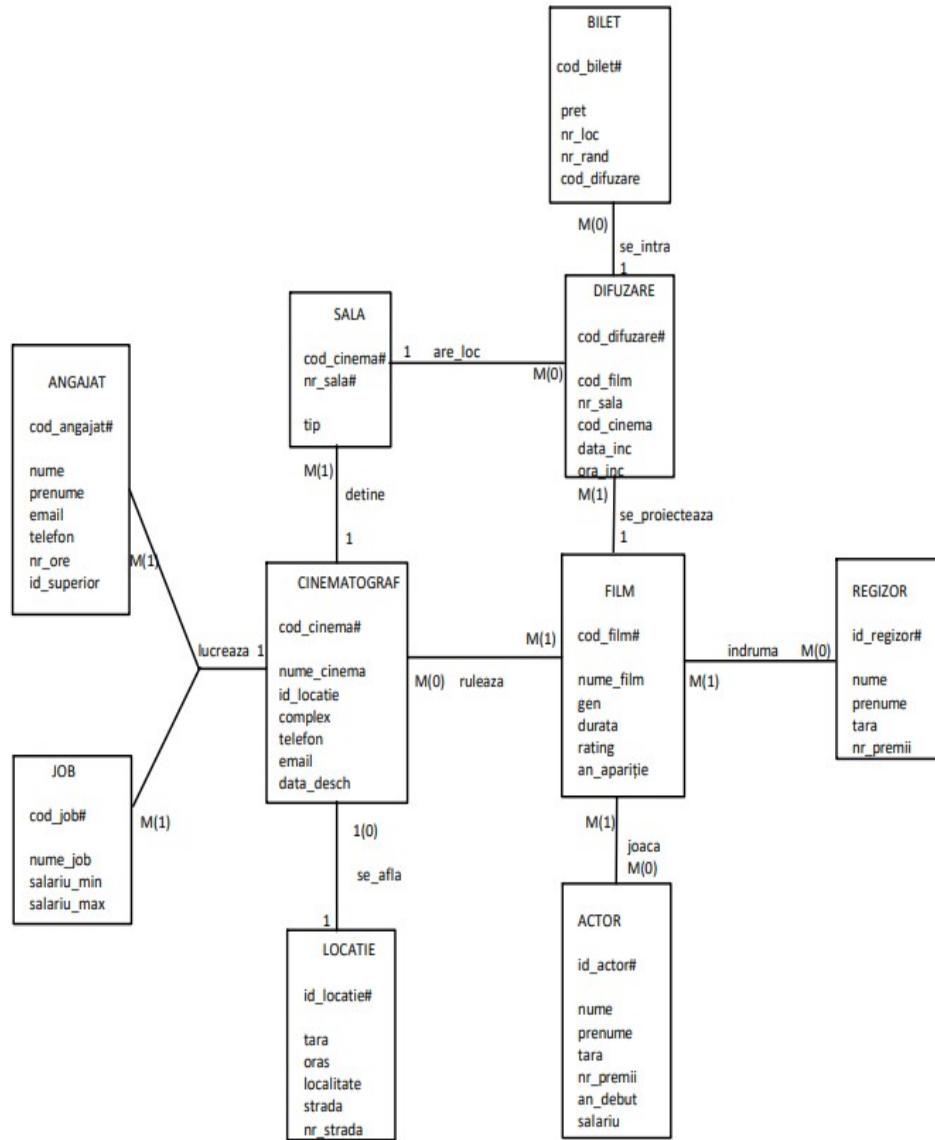
- a. **id_regizor** – numar de maxim 4 cifre (NUMBER(4)), care reprezinta id-ul unui regizor, acest id fiind unic. Aceasta reprezinta cheia primara a entitatii REGIZOR și nu poate avea valoarea „null”
- b. **nume** – sir de maxim 25 de caractere(litere mari, mici, „_”)(VARCHAR (25)), care precizeaza numele de familie al regizorului. Aceasta este diferit de „null”, deoarece trebuie să existe un nume după care să poată fi identificat

- c. **prenume** – sir de maxim 25 de caractere (litere mari, mici, „_”)(VARCHAR (25)), care precizeaza prenumele regizorului, fiind optionala precizarea acestuia
- d. **tara** – sir de maxim de 3 caractere (VARCHAR(3)), format din literele reprezentative ce alcătuiesc numele ţării respective, reprezentând id-ul ţării natale a regizorului respectiv (exemplu: RO). Aceasta este unic și diferit de „null”, deoarece orice persoană trebuie să aibă o țară natală
- e. **nr_premii** – număr de maxim 3 cifre (NUMBER(3)), care reprezintă numărul de premii importante (Oscar, Gopo etc) primite de regizorul respectiv până în prezent. Are valoare implicită 0 și este mai mare sau egal decât 0

6. Diagrama Entitate-Relație

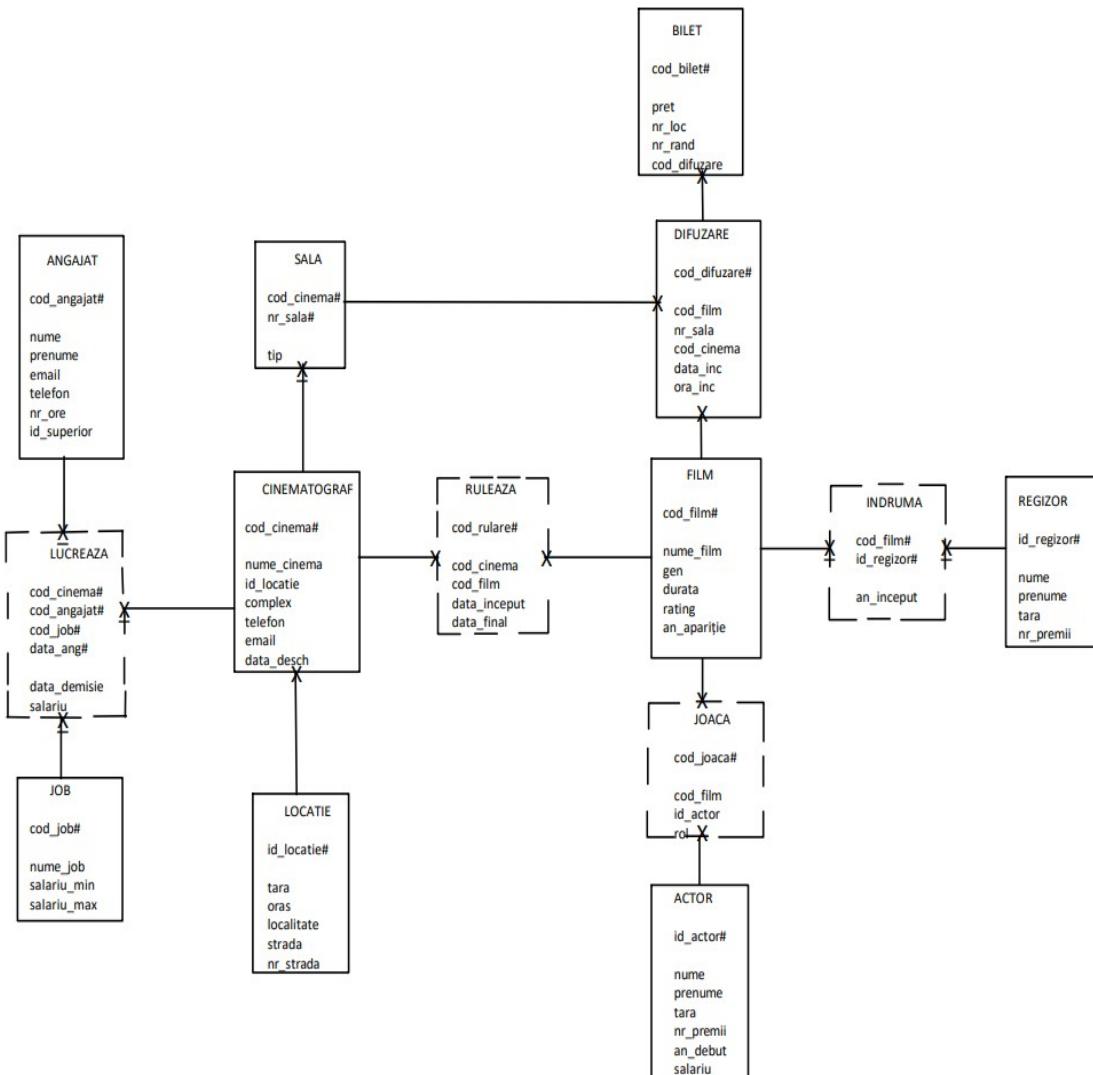
Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.

Diagrama Entitate-Relație. (proiectată în One Note)



7. Diagrama Conceptuală

Diagrama conceptuală corespunzătoare diagramei entitate-relație este prezentată în figura de mai jos.



8. Schemele relaționale

Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7.

Schemele relaționale corespunzătoare diagramei conceptuale sunt:

CINEMATOGRAF(cod_cinema#, nume_cinema, id_locatie, complex, telefon, email, data_desch)

SALA(cod_cinema#, nr_sala#, tip)

RULEAZA(cod_rulare#, cod_cinema#, cod_film#, data_ineput, data_final)

FILM(cod_film#, nume_film, gen, durata, rating, an_aparitie)

DIFUZARE(cod_difuzare#, cod_film, nr_sala, cod_cinema, data_inc, ora_inc)

BILET(cod_bilet#, pret, nr_loc, nr_rand, cod_difuzare)

REGIZOR(id_regizor#, nume, prenume, tara, nr_premii)

INDRUMA(cod_film#, id_regizor#, an_inceput)

ACTOR(id_actor#, nume, prenume, tara, nr_premii, an_debut, salariu)

JOACA(cod_joaca#, cod_film, id_actor, rol)

ANGAJAT(cod_angajat#, nume, prenume, email, telefon, nr_ore, id_superior)

LUCREAZA(cod_cinema#, cod_angajat#, cod_job#, data_ang#, data_demisie, salariu)

JOB(cod_job#, nume_job, salariu_min, salariu_max)

LOCATIE(id_locatie#, tara, oras, localitate, strada, nr_strada)

9. Realizarea normalizării până la forma normală 3 (FN1-FN3)

1. Exemplu de non-FN1 și transformarea acestuia în FN1:

Presupunem că entitatea FILM ar fi avut, inițial, următoarea schemă relațională:

FILM(cod_film#, nume_film, gen, durata, rating, an_aparitie, id_actor, nume, prenume, tara, nr_premii, an_debut, salariu, rol)

În cazul în care într-un film joacă mai mulți actori, atunci vom întâlni grupurile repetitive: „id_actor”, „nume”, „prenume”, „nr_premii”, „an_debut”, „salariu”, „rol”.

Deoarece într-un film pot juca mai mulți actori, atunci informațiile legate de distribuția filmelor vor apărea într-o altă tabelă. Aplicând etapele de aducere la FN1, se obțin două relații, cu schemele relaționale următoare:

FILM(cod_film#, nume_film, an_aparitie, rating)

JOACA(cod_film#, id_actor#, nume, prenume, tara, nr_premii, an_debut, salariu, rol)

2. Exemplu de non-FN2 și transformarea acestuia în FN2:

Presupunem că relația JOACA ar fi avut, inițial, următoarea schemă relațională:

JOACA(cod_film#, id_actor#, nume, prenume, tara, nr_premii, an_debut, salariu, rol)

Se observă că atributele „nume”, „prenume”, „tara”, „nr_premii”, „an_debut”, „salariu”, „rol” nu depind de întreaga cheie primară, ci doar de „id_actor”. Relația este în FN1, dar nu și în FN2. Aplicând etapele de aducere la FN2, se obțin două relații, cu schemele relaționale următoare:

JOACA(cod_film#, id_actor#, rol)

ACTOR(id_actor#, nume, prenume, tara, nr_premii, an_debut, salariu)

3. Exemplu de non-FN3 și transformarea acestuia în FN3:

Presupunem că entitatea CINEMATOGRAF ar fi avut, inițial, următoarea schemă relațională:

CINEMATOGRAF(cod_cinema#,nume_cinema, id_locatie,complex,telefon,email,
data_desch, tara, oras, localitate, strada, nr_strada)

Relația este în FN2, dar nu este în FN3, deoarece atributele „tara”, „oras”, „strada” și „nr_strada” depind indirect de cheia primară, prin intermediul atributului „id_locatie”.

Aplicând etapele de aducere la FN3, se obțin două relații, cu schemele relaționale următoare:

CINEMATOGRAF(cod_cinema#,nume_cinema, id_locatie,complex,telefon,email,
data_desch)

LOCATIE(id_locatie#, tara, oras,localitate, strada, nr_strada)

10. Crearea secvențelor

Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11) este prezentată în continuare.

Pentru tabela **LOCATIE** voi implementa o secvență care să îmi genereze id-ul locației:

```
create sequence seq_id_locatie
increment by 5
start with 100
maxvalue 9999
nocycle
cache 10;
```

```
create sequence seq_id_locatie
increment by 5
start with 100
maxvalue 9999
nocycle
cache 10;
```

Script Output | Task completed in 0.025 seconds

Sequence SEQ_ID_LOCATIE created.

Pentru tabela **CINEMATOGRAF** voi implementa o secvență care să îmi genereze codul fiecărui cinematograf:

```
create sequence seq_cod_cinema
increment by 20
start with 100
maxvalue 9999
nocycle
cache 10;
```

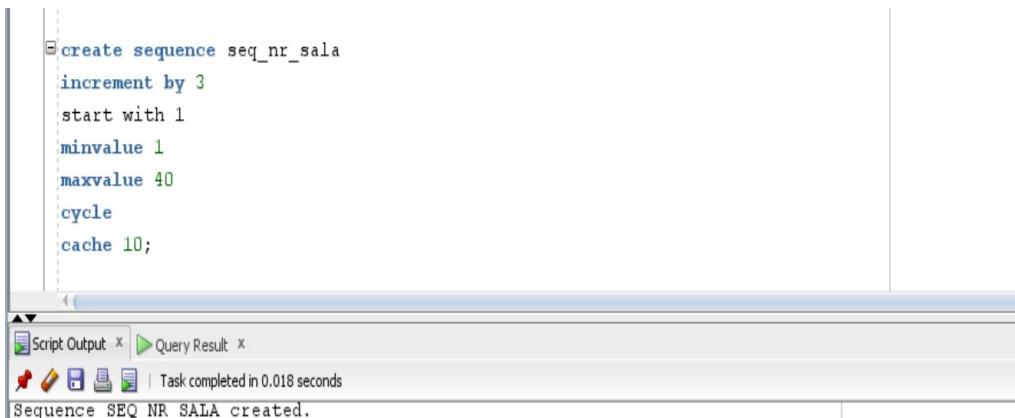
```
create sequence seq_cod_cinema
increment by 20
start with 100
maxvalue 9999
nocycle
cache 10;
```

Script Output | Task completed in 0.027 seconds

Sequence SEQ_COD_CINEMA created.

Pentru tabela **SALA** voi implementa o secvență care să îmi genereze numerele sălilor:

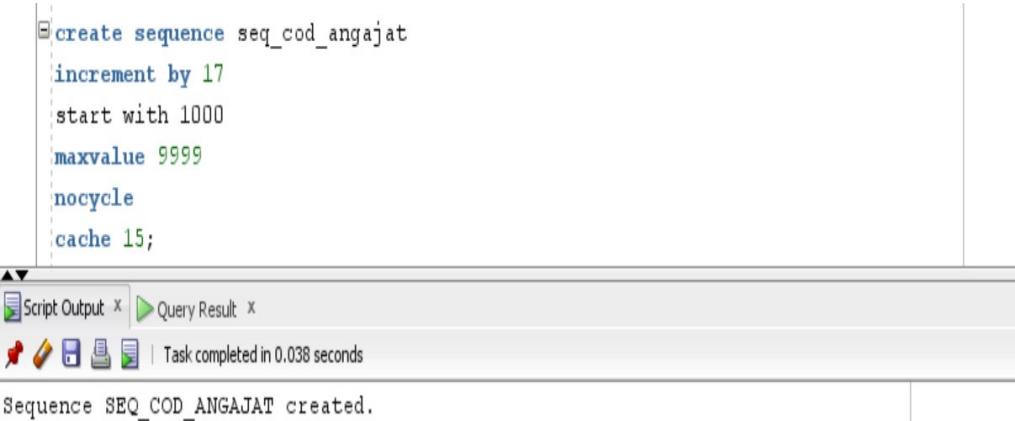
```
create sequence seq_nr_sala
increment by 3
start with 1
minvalue 1
maxvalue 40
cycle
cache 10;
```



The screenshot shows the Oracle SQL Developer interface. A code editor window contains the SQL command to create a sequence named seq_nr_sala with specific parameters. Below the editor, a toolbar has icons for script, edit, and query. A status bar at the bottom indicates 'Task completed in 0.018 seconds' and shows the message 'Sequence SEQ_NR_SALA created.'

Pentru tabela **ANGAJAT** voi implementa o secvență care să îmi genereze codul fiecărui angajat:

```
create sequence seq_cod_angajat
increment by 17
start with 1000
maxvalue 9999
nocycle
cache 15;
```



The screenshot shows the Oracle SQL Developer interface. A code editor window contains the SQL command to create a sequence named seq_cod_angajat with specific parameters. Below the editor, a toolbar has icons for script, edit, and query. A status bar at the bottom indicates 'Task completed in 0.038 seconds' and shows the message 'Sequence SEQ_COD_ANGAJAT created.'

Pentru tabela **FILM** voi implementa o secvență care să îmi genereze codul fiecărui film:

```
create sequence seq_cod_film
increment by 100
start with 10
maxvalue 9999
nocycle
cache 20;
```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor window containing the SQL command to create a sequence. Below the code editor is a toolbar with icons for script, output, and query result. The status bar at the bottom indicates "Task completed in 0.024 seconds". In the bottom-right pane, the output window displays the message "Sequence SEQ_COD_FILM created.".

```
create sequence seq_cod_film
increment by 100
start with 10
maxvalue 9999
nocycle
cache 20;
```

Sequence SEQ_COD_FILM created.

Pentru tabela **DIFUZARE** voi implementa o secvență care să îmi genereze codul fiecărei difuzari:

```
create sequence seq_cod_difuzare
increment by 21
start with 50
maxvalue 9999
nocycle
cache 25;
```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor window containing the SQL command to create a sequence. Below the code editor is a toolbar with icons for script, output, and query result. The status bar at the bottom indicates "Task completed in 0.025 seconds". In the bottom-right pane, the output window displays the message "Sequence SEQ_COD_DIFUZARE created.".

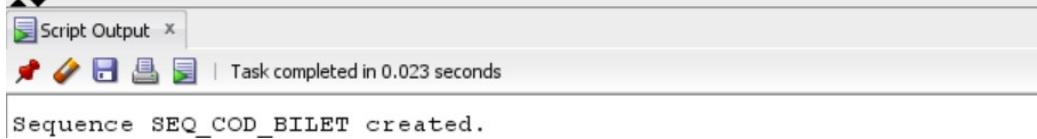
```
create sequence seq_cod_difuzare
increment by 21
start with 50
maxvalue 9999
nocycle
cache 25;
```

Sequence SEQ_COD_DIFUZARE created.

Pentru tabela **BILET** voi implementa o secvență care să îmi genereze codul fiecărui bilet:

```
create sequence seq_cod_bilet
increment by 5
start with 1
maxvalue 9999
nocycle
cache 10;
```

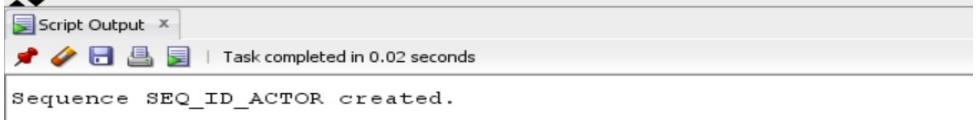
```
create sequence seq_cod_bilet
increment by 5
start with 1
maxvalue 9999
nocycle
cache 10;
```



Pentru tabela **ACTOR** voi implementa o secvență care să îmi genereze id-ul fiecărui actor:

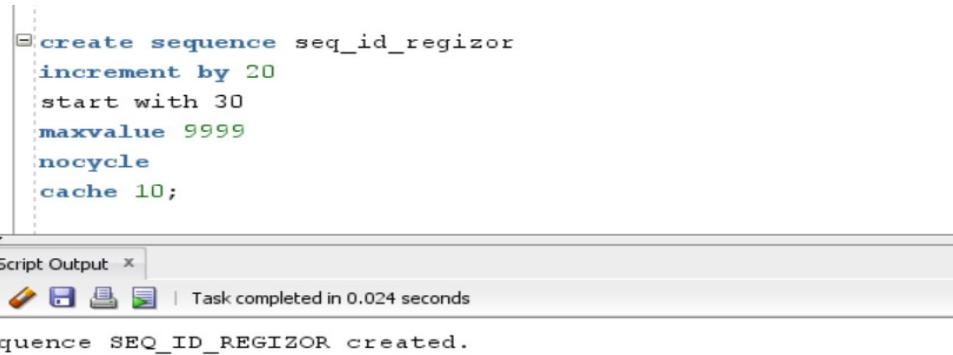
```
create sequence seq_id_actor
increment by 15
start with 10
maxvalue 9999
nocycle
cache 10;
```

```
create sequence seq_id_actor
increment by 15
start with 10
maxvalue 9999
nocycle
cache 10;
```



Pentru tabela **REGIZOR** voi implementa o secvență care să îmi genereze id-ul fiecărui regizor:

```
create sequence seq_id_regizor
increment by 20
start with 30
maxvalue 9999
nocycle
cache 10;
```



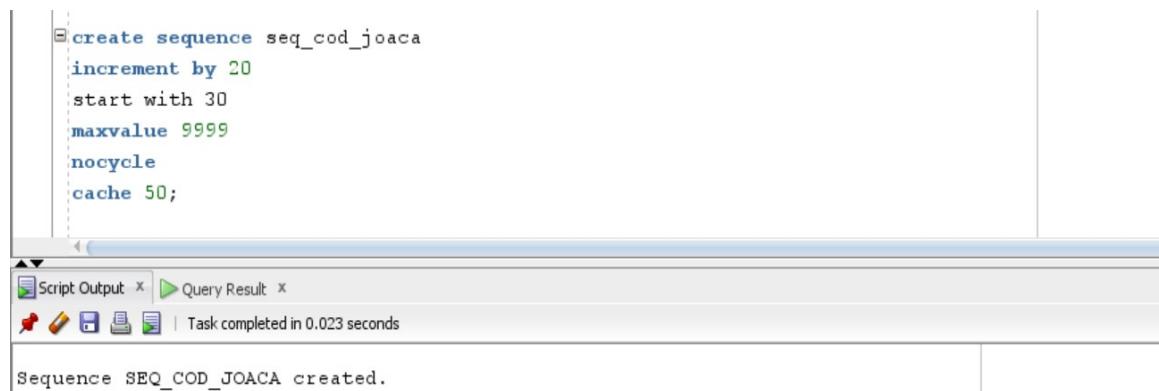
The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor window containing the SQL command to create a sequence. Below the code editor is a toolbar with various icons. At the bottom of the interface, there is a status bar displaying the message "Task completed in 0.024 seconds". The status bar also includes a "Script Output" tab and a "Query Result" tab.

```
create sequence seq_id_regizor
increment by 20
start with 30
maxvalue 9999
nocycle
cache 10;
```

Sequence SEQ_ID_REGIZOR created.

Pentru tabela **JOACA** voi implementa o secvență care să îmi genereze codul atribuit filmului în care joacă un actor, al actorului și rolul din film:

```
create sequence seq_cod_joaca
increment by 20
start with 30
maxvalue 9999
nocycle
cache 50;
```



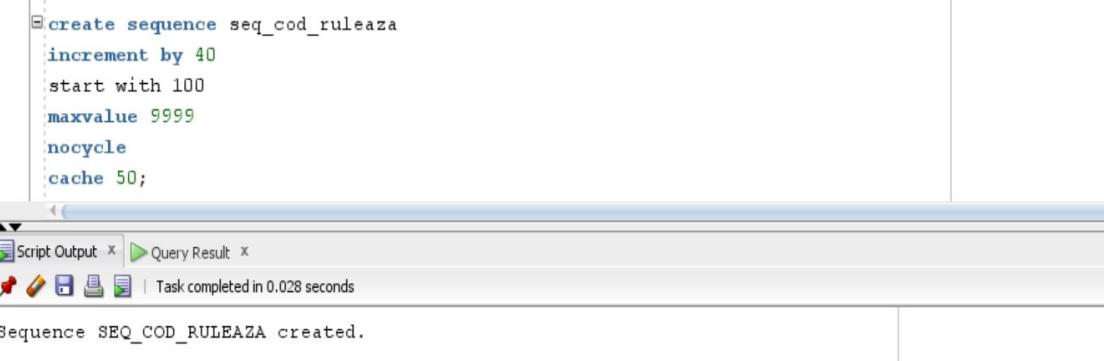
The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor window containing the SQL command to create a sequence. Below the code editor is a toolbar with various icons. At the bottom of the interface, there is a status bar displaying the message "Task completed in 0.023 seconds". The status bar also includes a "Script Output" tab and a "Query Result" tab.

```
create sequence seq_cod_joaca
increment by 20
start with 30
maxvalue 9999
nocycle
cache 50;
```

Sequence SEQ_COD_JOACA created.

Pentru tabela **RULEAZA** voi implementa o secvență care să îmi genereze codul atribuit filmului care rulează într-un cinematograf, ținându-se cont de faptul că un film poate rula după un interval de timp din nou:

```
create sequence seq_cod_ruleaza
increment by 40
start with 100
maxvalue 9999
nocycle
cache 50;
```



The screenshot shows the Oracle SQL Developer interface. A code editor window contains the SQL command to create a sequence named 'seq_cod_ruleaza' with specific parameters. Below the editor, a message indicates the sequence was successfully created. The interface also shows tabs for 'Script Output' and 'Query Result', and a status bar at the bottom.

```
create sequence seq_cod_ruleaza
increment by 40
start with 100
maxvalue 9999
nocycle
cache 50;
```

Sequence SEQ_COD_RULEAZA created.

11. Crearea tabelelor, inserarea datelor

Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative).

Crearea tabelei **LOCATIE**:

```
create table locatie(
    id_locatie number(4) constraint pk_locatie primary key,
    tara varchar(3) constraint null_tara_loc not null,
    oras varchar(20) constraint null_oras_loc not null,
    localitate varchar(20),
    strada varchar(20) constraint null_strada_loc not null,
    nr_strada number(2));
```

```
create table locatie(
    id_locatie number(4) constraint pk_locatie primary key,
    tara varchar(3) constraint null_tara_loc not null,
    oras varchar(20) constraint null_oras_loc not null,
    localitate varchar(20),
    strada varchar(20) constraint null_strada_loc not null,
    nr_strada number(2));
```



Crearea tabelei **JOB**:

```
create table job(
    cod_job varchar(6) constraint pk_job primary key,
    nume_job varchar(20) constraint null_nume_job not null,
    salariu_min number(8, 2) constraint null_sal_min not null,
    salariu_max number(8, 2) constraint null_sal_max not null,
    constraint unq_nume_job unique(nume_job),
    constraint ck_sal_min check(salariu_min > 0.0),
    constraint ck_sal_max check(salariu_max > 0.0));
```

```

create table job(
    cod_job varchar(6) constraint pk_job primary key,
    nume_job varchar(20) constraint null_nume_job not null,
    salariu_min number(8, 2) constraint null_sal_min not null,
    salariu_max number(8, 2) constraint null_sal_max not null,
    constraint unq_nume_job unique(nume_job),
    constraint ck_sal_min check(salariu_min > 0.0),
    constraint ck_sal_max check(salariu_max > 0.0));

```

Crearea tebelei ANGAJAT:

```

create table angajat(
    cod_angajat number(4) constraint pk_angajat primary key,
    nume varchar(25) constraint null_nume_ang not null,
    prenume varchar(25),
    email char(25) constraint null_email_ang not null,
    telefon varchar2(20),
    nr_ore number(2) constraint null_nr_ore not null,
    id_superior number(4) constraint fk_ang_ang_sup references
angajat(cod_angajat) on delete cascade,
    constraint unq_nume_pren_ang unique(nume, prenume),
    constraint unq_email_ang unique(email),
    constraint ck_nr_ore_ang check(nr_ore > 0 and nr_ore < 24));

```

```

create table angajat(
    cod_angajat number(4) constraint pk_angajat primary key,
    nume varchar(25) constraint null_nume_ang not null,
    prenume varchar(25),
    email char(25) constraint null_email_ang not null,
    telefon varchar2(20),
    nr_ore number(2) constraint null_nr_ore not null,
    id_superior number(4) constraint fk_ang_ang_sup references angajat(cod_angajat) on delete cascade,
    constraint unq_nume_pren_ang unique(nume, prenume),
    constraint unq_email_ang unique(email),
    constraint ck_nr_ore_ang check(nr_ore > 0 and nr_ore < 24));

```

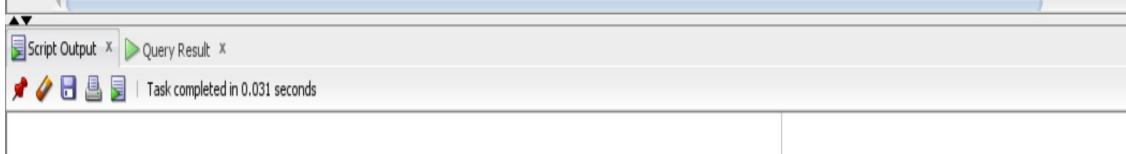


Table ANGAJAT created.

Crearea tabelei CINEMATOGRAF:

```

create table cinematograf(
    cod_cinema number(4) constraint pk_cinema primary key,
    nume_cinema varchar(25) constraint null_nume_cinema not null,
    id_locatie number(4) constraint null_loc_cinema not null,
    complex varchar2(30),
    telefon varchar2(20),
    email char(25) constraint null_email_cinema not null,
    data_desch date default sysdate,
    constraint unq_email_cinema unique(email),
    constraint fk_cinem_loc foreign key(id_locatie) references locatie(id_locatie) on
    delete cascade);
    alter table cinematograf add constraint unqid_loc_cin unique(id_locatie);

```

```

create table cinematograf(
    cod_cinema number(4) constraint pk_cinema primary key,
    nume_cinema varchar(25) constraint null_nume_cinema not null,
    id_locatie number(4) constraint null_loc_cinema not null,
    complex varchar2(30),
    telefon varchar2(20),
    email char(25) constraint null_email_cinema not null,
    data_desch date default sysdate,
    constraint unq_email_cinema unique(email),
    constraint fk_cinem_loc foreign key(id_locatie) references locatie(id_locatie) on delete cascade);

```



Table CINEMATOGRAF created.

```
| alter table cinematograf add constraint unqid_loc_cin unique(id_locatie);
```

Script Output X | Query Result X
Table CINEMATOGRAPF altered.

Crearea tablei asociative **LUCREAZA**:

```
create table lucreaza(
    cod_cinema number(4) constraint fk_cin_lucr references
    cinematograf(cod_cinema),
    cod_angajat number(4)constraint fk_ang_lucr references angajat(cod_angajat),
    cod_job varchar(6)constraint fk_job_lucr references job(cod_job),
    data_ang date default sysdate,
    data_demisie date,
    salariu number(8,2),
    constraint pk_ang_lucr_cin primary key( cod_angajat, cod_cinema, cod_job,
    data_ang),
    constraint ck_sal_ang check(salariu > 0));
```

```
create table lucreaza(
    cod_cinema number(4) constraint fk_cin_lucr references cinematograf(cod_cinema),
    cod_angajat number(4)constraint fk_ang_lucr references angajat(cod_angajat),
    cod_job varchar(6)constraint fk_job_lucr references job(cod_job),
    data_ang date default sysdate,
    data_demisie date,
    salariu number(8,2) constraint null_salariu_ang not null,
    constraint pk_ang_lucr_cin primary key(cod_cinema, cod_angajat, cod_job, data_ang),
    constraint ck_sal_ang check(salariu > 0));
```

Script Output X | Query Result X
Table LUCREAZA created.

Crearea tablei **FILM**:

```
create table film(
    cod_film number(4) constraint pk_film primary key,
    nume_film varchar(25) constraint null_nume_film not null,
    gen varchar(20) constraint null_gen not null,
```

```

durata number(3) default 0,
rating number(2, 1) constraint ck_rating check(rating > 0.0 and rating <= 10.0),
an_aparitie number(4) constraint null_an_aparitie not null,
constraint ck_an_ap check(an_aparitie > 0),
constraint unq_numefilm_an_ap unique(numefilm, an_aparitie));

create table film(
cod_film number(4) constraint pk_film primary key,
numefilm varchar(25) constraint null_numefilm not null,
gen varchar(20) constraint null_gen not null,
durata number(3) default 0,
rating number(2, 1) constraint ck_rating check(rating > 0.0 and rating <= 10.0),
an_aparitie number(4) constraint null_an_aparitie not null,
constraint ck_an_ap check(an_aparitie > 0),
constraint unq_numefilm_an_ap unique(numefilm, an_aparitie));

```

Script Output X | Query Result X
✖ ✔ ✖ ✖ ✖ | Task completed in 0.032 seconds

Table FILM created.

Crearea tablei asociative RULEAZA:

```

create table ruleaza(
cod_rulare number(4) constraint pk_ruleaza primary key,
cod_cinema number(4) constraint fk_cin_rul references
cinematograf(cod_cinema),
cod_film number(4) constraint fk_film_rul references film(cod_film),
data_inceput date default sysdate,
data_final date default sysdate);

```

```

create table ruleaza(
cod_rulare number(4) constraint pk_ruleaza primary key,
cod_cinema number(4) constraint fk_cin_rul references cinematograf(cod_cinema),
cod_film number(4) constraint fk_film_rul references film(cod_film),
data_inceput date default sysdate,
data_final date default sysdate);

```

Script Output X | Query Result X
✖ ✔ ✖ ✖ ✖ | Task completed in 0.038 seconds

Table RULEAZA created.

Crearea tablei ACTOR:

```
create table actor(
    id_actor number(4) constraint pk_actor primary key,
    prenume varchar(25),
    nume varchar(25) constraint null_nume_actor not null,
    tara varchar(3) constraint null_tara_act not null,
    nr_premii number(3) default 0,
    an_debut number(4) constraint null_an_debut_act not null,
    salariu number(8, 2) constraint null_sal_act not null,
    constraint unq_nume_pren_act unique(nume, prenume),
    constraint ck_nr_premii_act check(nr_premii >= 0),
    constraint ck_an_debut_act check(an_debut > 0),
    constraint ck_sal_act check(salariu > 0));
```

The screenshot shows the Oracle SQL Developer interface. A code editor window contains the SQL script for creating the ACTOR table. Below the editor, a toolbar has icons for script, edit, run, and refresh. A status bar at the bottom indicates "Task completed in 0.029 seconds". A message in the status bar says "Table ACTOR created."

```
create table actor(
    id_actor number(4) constraint pk_actor primary key,
    prenume varchar(25),
    nume varchar(25) constraint null_nume_actor not null,
    tara varchar(3) constraint null_tara_act not null,
    nr_premii number(3) default 0,
    an_debut number(4) constraint null_an_debut_act not null,
    salariu number(8, 2) constraint null_sal_act not null,
    constraint unq_nume_pren_act unique(nume, prenume),
    constraint ck_nr_premii_act check(nr_premii >= 0),
    constraint ck_an_debut_act check(an_debut > 0),
    constraint ck_sal_act check(salariu > 0));
```

Crearea tablei asociative JOACA:

```
create table joaca(
    cod_joaca number(4) constraint pk_joaca primary key,
    cod_film number(4) constraint fk_film_joaca references film(cod_film),
    id_actor number(4) constraint fk_act_joaca references actor(id_actor),
    rol varchar(20) constraint null_rol_joaca not null);
```

```

create table joaca(
    cod_joaca number(4) constraint pk_joaca primary key,
    cod_film number(4) constraint fk_film_joaca references film(cod_film),
    id_actor number(4) constraint fk_act_joaca references actor(id_actor),
    rol varchar(20) constraint null_rol_joaca not null);

```

```

create table joaca(
    cod_film number(4) constraint fk_film_joaca references film(cod_film),
    id_actor number(4) constraint fk_act_joaca references actor(id_actor),
    rol varchar(20) constraint null_rol_joaca not null,
    constraint pk_act_film primary key(cod_film, id_actor));

```

Script Output | Query Result | Task completed in 0.024 seconds

Table JOACA created.

Crearea tabelei REGIZOR:

```

create table regizor(
    id_regizor number(4) constraint pk_regizor primary key,
    prenume varchar(25),
    nume varchar(25) constraint null_nume_regizor not null,
    tara varchar(3) constraint null_tara_reg not null,
    nr_premii number(3) default 0,
    constraint unq_nume_pren_reg unique(nume, prenume),
    constraint ck_nr_premii_reg check(nr_premii >= 0));

```

```

create table regizor(
    id_regizor number(4) constraint pk_regizor primary key,
    prenume varchar(25),
    nume varchar(25) constraint null_nume_regizor not null,
    tara varchar(3) constraint null_tara_reg not null,
    nr_premii number(3) default 0,
    constraint unq_nume_pren_reg unique(nume, prenume),
    constraint ck_nr_premii_reg check(nr_premii >= 0));

```

Script Output | Query Result | Task completed in 0.033 seconds

Table REGIZOR created.

Crearea tableei asociative INDRUMA:

```
create table indruma(
    cod_film number(4) constraint fk_film_indruma references film(cod_film),
    id_regizor number(4) constraint fk_regizor_indruma references
    regizor(id_regizor),
    an_inceput number(4) constraint null_an_inc_indr not null,
    constraint pk_regizor_film primary key(cod_film, id_regizor),
    constraint ck_an_inc_indr check(an_inceput > 0));
```

```
create table indruma(
    cod_film number(4) constraint fk_film_indruma references film(cod_film),
    id_regizor number(4) constraint fk_regizor_indruma references regizor(id_regizor),
    an_inceput number(4) constraint null_an_inc_indr not null,
    constraint pk_regizor_film primary key(cod_film, id_regizor),
    constraint ck_an_inc_indr check(an_inceput > 0));
```

Script Output | Query Result | Task completed in 0.034 seconds

Table INDRUMA created.

Crearea tableei SALA:

```
create table sala(
    nr_sala number(2),
    cod_cinema number(4),
    tip varchar(10),
    constraint ck_nr_sala check(nr_sala <= 40 and nr_sala > 0),
    constraint fk_sala_cinema foreign key(cod_cinema) references
    cinematograf(cod_cinema),
    constraint pk_sala_cin primary key(nr_sala, cod_cinema));
```

```
create table sala(
    nr_sala number(2),
    cod_cinema number(4),
    tip varchar(10),
    constraint ck_nr_sala check(nr_sala <= 40 and nr_sala > 0),
    constraint fk_sala_cinema foreign key(cod_cinema) references cinematograf(cod_cinema),
    constraint pk_sala_cin primary key(nr_sala, cod_cinema));
```

Script Output | Query Result | Task completed in 0.031 seconds

Table SALA created.

Crearea tabelei DIFUZARE:

```
create table difuzare(
    cod_difuzare number(4) constraint pk_difuzare primary key,
    cod_film number(4) constraint null_dif_film not null,
    nr_sala number(2) constraint null_sala_dif not null,
    cod_cinema number(4) constraint null_cin_dif not null,
    data_inc date default sysdate,
    ora_inc number(2) constraint null_ora_inc not null,
    constraint ck_ora_inc check(ora_inc > 0 and ora_inc < 24),
    constraint fk_film_dif foreign key(cod_film) references film(cod_film),
    constraint fk_nr_sala_cin_dif foreign key(nr_sala, cod_cinema) references
    sala(nr_sala, cod_cinema));
```

The screenshot shows the Oracle SQL Developer interface with the SQL worksheet tab active. The code for creating the DIFUZARE table is pasted into the worksheet. Below the code, the status bar indicates "Task completed in 0.034 seconds". The message "Table DIFUZARE created." is displayed at the bottom of the worksheet area.

```
create table difuzare(
    cod_difuzare number(4) constraint pk_difuzare primary key,
    cod_film number(4) constraint null_dif_film not null,
    nr_sala number(2) constraint null_sala_dif not null,
    cod_cinema number(4) constraint null_cin_dif not null,
    data_inc date default sysdate,
    ora_inc number(2) constraint null_ora_inc not null,
    constraint ck_ora_inc check(ora_inc > 0 and ora_inc < 24),
    constraint fk_film_dif foreign key(cod_film) references film(cod_film),
    constraint fk_nr_sala_cin_dif foreign key(nr_sala, cod_cinema) references
    sala(nr_sala, cod_cinema));
```

Crearea tabelei BILET:

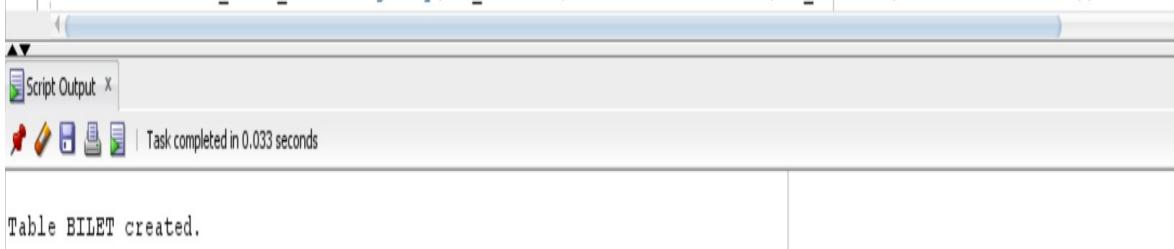
```
create table bilet(
    cod_bilet number(4) constraint pk_bilet primary key,
    pret number(8.2) default 0.0,
    nr_loc number(3) constraint null_loc_bilet not null,
    nr_rand number(3) constraint null_rand_bilet not null,
    cod_difuzare number(4) constraint null_cod_dif_bilet not null,
    constraint ck_pret check(pret >= 0.0),
    constraint ck_nr_loc check(nr_loc > 0 and nr_loc <= 300),
    constraint ck_nr_rand check(nr_rand > 0 and nr_rand <= 20),
```

```

constraint fk_bilet_dif foreign key(cod_difuzare) references
difuzare(cod_difuzare) on delete cascade);

create table bilet(
    cod_bilet number(4) constraint pk_bilet primary key,
    pret number(8, 2) default 0.0,
    nr_loc number(3) constraint null_loc_bilet not null,
    nr_rand number(3) constraint null_rand_bilet not null,
    cod_difuzare number(4) constraint null_cod_dif_bilet not null,
    constraint ck_pret check(pret >= 0.0),
    constraint ck_nr_loc check(nr_loc > 0 and nr_loc <= 300),
    constraint ck_nr_rand check(nr_rand > 0 and nr_rand <= 20),
    constraint fk_bilet_dif foreign key(cod_difuzare) references difuzare(cod_difuzare) on delete cascade);

```



Inserări în tabela LOCATIE:

```

insert into locatie
values(seq_id_locatie.nextval, 'RO', 'Bucuresti', 'Bucuresti', 'Liviu Rebreanu', 4);

insert into locatie
values(seq_id_locatie.nextval, 'RO', 'Bucuresti', 'Bucuresti', 'Bd Timisoara', 26);

insert into locatie
values(seq_id_locatie.nextval, 'RO', 'Bucuresti', 'Bucuresti', 'Mihai Eminescu', null);

insert into locatie
values(seq_id_locatie.nextval, 'RO', 'Sibiu', null, 'Aurel Vlaicu', 11);

insert into locatie
values(seq_id_locatie.nextval, 'RO', 'Suceava', 'Suceava', 'Energeticului', null);

insert into locatie
values(seq_id_locatie.nextval, 'RO', 'Iasi', 'Iasi', 'Piata Unirii', 5);

insert into locatie

```

```

values(seq_id_locatie.nextval, 'RO', 'Constanta', null, 'Aurel Vlaicu', 2);

insert into locatie
values(seq_id_locatie.nextval, 'RO', 'Cluj-Napoca', null, 'Bd Eroilor', 51);

insert into locatie
values(seq_id_locatie.nextval, 'RO', 'Baia Mare', null, 'Victoriei', 73);

insert into locatie
values(seq_id_locatie.nextval, 'RO', 'Pitesti', null, 'Justitiei', 1);

insert into locatie
values(seq_id_locatie.nextval, 'RO', 'Bucuresti', 'Bucuresti', 'Liviu Rebreanu', 4);
insert into locatie
values(seq_id_locatie.nextval, 'RO', 'Bucuresti', 'Bucuresti', 'Bd Timisoara', 26);
insert into locatie
values(seq_id_locatie.nextval, 'RO', 'Bucuresti', 'Bucuresti', 'Mihai Eminescu', null);
insert into locatie
values(seq_id_locatie.nextval, 'RO', 'Sibiu', null, 'Aurel Vlaicu', 11);
insert into locatie
values(seq_id_locatie.nextval, 'RO', 'Suceava', 'Suceava', 'Energeticului', null);
insert into locatie
values(seq_id_locatie.nextval, 'RO', 'Iasi', 'Iasi', 'Piata Unirii', 5);
insert into locatie
values(seq_id_locatie.nextval, 'RO', 'Constanta', null, 'Aurel Vlaicu', 2);
insert into locatie
values(seq_id_locatie.nextval, 'RO', 'Cluj-Napoca', null, 'Bd Eroilor', 51);
insert into locatie
values(seq_id_locatie.nextval, 'RO', 'Baia Mare', null, 'Victoriei', 73);
insert into locatie
values(seq_id_locatie.nextval, 'RO', 'Pitesti', null, 'Justitiei', 1);

```

```
select * from locatie;
```

Script Output X Query Result X

SQL | All Rows Fetched: 10 in 0.006 seconds

ID_LOCATIE	TARA	ORAS	LOCALITATE	STRADA	NR_STRADA
1	105 RO	Bucuresti	Bucuresti	Liviu Rebreanu	4
2	110 RO	Bucuresti	Bucuresti	Bd Timisoara	26
3	115 RO	Bucuresti	Bucuresti	Mihai Eminescu	(null)
4	120 RO	Sibiu	(null)	Aurel Vlaicu	11
5	125 RO	Suceava	Suceava	Energeticului	(null)
6	130 RO	Iasi	Iasi	Piata Unirii	5
7	135 RO	Constanta	(null)	Aurel Vlaicu	2
8	140 RO	Cluj-Napoca	(null)	Bd Eroilor	51
9	145 RO	Baia Mare	(null)	Victoriei	73
10	150 RO	Pitesti	(null)	Justitiei	1

Inserări în tabela **JOB**:

```
insert into job
values('CAS', 'Casier', 3792.1, 6293.0);

insert into job
values('AG_C', 'Agent de Curatenie', 1239.2, 999.9);

insert into job
values('MAN', 'Manager', 5439.22, 6934.2);

insert into job
values('ING_S', 'Inginer Sunet', 3234.2, 4593.3);

insert into job
values('ING_T', 'Inginer Tehnic', 2134.2, 5593.3);

insert into job
values('PLAS', 'Plasator', 1231.4, 3432.43);

insert into job
values('BAR_C', 'Barman Cafenea', 1312.3, 2313.3);

insert into job
values('BAR', 'Barman', 1192.3, 1922.0);
```

```
insert into job
values('CAS', 'Casier', 3792.1, 6293.0);
insert into job
values('AG_C', 'Agent de Curatenie', 1239.2, 999.9);
insert into job
values('MAN', 'Manager', 5439.22, 6934.2);
insert into job
values('ING_S', 'Inginer Sunet', 3234.2, 4593.3);
insert into job
values('ING_T', 'Inginer Tehnic', 2134.2, 5593.3);
insert into job
values('PLAS', 'Plasator', 1231.4, 3432.43);
insert into job
values('BAR_C', 'Barman Cafenea', 1312.3, 2313.3);
insert into job
values('BAR', 'Barman', 1192.3, 1922.0);
```

The screenshot shows the Oracle SQL Developer interface. In the top-left, there is a code editor window containing the SQL command:

```
select * from job;
```

Below the code editor is a toolbar with icons for Script Output, Query Result, and other database operations. The status bar indicates "All Rows Fetched: 8 in 0.025 seconds".

The main area displays the results of the query in a grid format:

COD_JOB	NUME_JOB	SALARIU_MIN	SALARIU_MAX
1 CAS	Casier	3792.1	6293
2 ING_S	Inginer Sunet	3234.2	4593.3
3 ING_T	Inginer Tehnic	2134.2	5593.3
4 AG_C	Agent de Curatenie	1239.2	999.9
5 MAN	Manager	5439.22	6934.2
6 PLAS	Plasator	1231.4	3432.43
7 BAR	Barman	1192.3	1922
8 BAR_C	Barman Cafenea	1312.3	2313.3

Inserări în tabela ANGAJAT:

```
insert into angajat
values(seq_cod_angajat.nextval, 'Popescu', 'Ion', 'ion.popescu@gmail.com',
'0741.311.314', 8, null);

insert into angajat
values(seq_cod_angajat.nextval, 'Georgesc', 'Vasile', 'vasile.georgesc@gmail.com', null,
8, 1000);

insert into angajat
values(seq_cod_angajat.nextval, 'Matei', 'Andrei', 'andrei.matei@gmail.com',
'0799.532.678', 6, 1000);

insert into angajat
values(seq_cod_angajat.nextval, 'Iana', 'Gabriela', 'gabriela.iana@gmail.com',
'0712.623.881', 6, 1034);

insert into angajat
values(seq_cod_angajat.nextval, 'Tudor', 'Maria', 'maria.tudor@gmail.com',
'0712.133.292', 10, null);

insert into angajat
values(seq_cod_angajat.nextval, 'Miron', 'Costin', 'costin.miron@gmail.com',
'0771.229.345', 7, 1068);

insert into angajat
values(seq_cod_angajat.nextval, 'Banu', 'Codrin', 'codrin.banu@gmail.com',
'0741.721.921', 9, null);

insert into angajat
```

```

values(seq_cod_angajat.nextval, 'Crivat', 'Gheorghe', 'gheo.crivat@gmail.com', null, 8,
1102);

insert into angajat
values(seq_cod_angajat.nextval, 'Stan', 'Elena', 'elena.stan@gmail.com', '0789.123.433',
10, null);

insert into angajat
values(seq_cod_angajat.nextval, 'Pitco', 'Natanael', 'nate.pitco@gmail.com', null, 7,
1136);

insert into angajat
values(seq_cod_angajat.nextval, 'Florea', 'Ana', 'ana.florea@gmail.com', null, 8, 1136);

insert into angajat
values(seq_cod_angajat.nextval, 'Nistor', 'Viviana', 'viv.nistor@gmail.com',
'0712.385.763', 6, null);

insert into angajat
values(seq_cod_angajat.nextval, 'Leona', 'Antonia', 'anto.leona@gmail.com',
'0711.332.454', 12, 1187);

insert into angajat
values(seq_cod_angajat.nextval, 'Nils', 'Corina', 'corina.nils@gmail.com', null, 8, 1187);

insert into angajat
values(seq_cod_angajat.nextval, 'Petre', 'Cosmin', 'cosmin.petre@gmail.com',
'0718.223.344', 10, null);

insert into angajat
values(seq_cod_angajat.nextval, 'Popa', 'Maria', 'maria.popa@yahoo.com',
'0792.233.211', 8, null);

insert into angajat
values(seq_cod_angajat.nextval, 'Dumitran', 'George', 'george.dumi@gmail.com',
'0793.557.876', 9, null);

insert into angajat
values(seq_cod_angajat.nextval, 'Popescu', 'Ion', 'ion.popescu@gmail.com', '0741.311.314', 8, null);
insert into angajat
values(seq_cod_angajat.nextval, 'Georgesc', 'Vasile', 'vasile.georgesc@gmail.com', null, 8, 1000);
insert into angajat
values(seq_cod_angajat.nextval, 'Matei', 'Andrei', 'andrei.matei@gmail.com', '0799.532.678', 6, 1000);
insert into angajat
values(seq_cod_angajat.nextval, 'Tana', 'Gabriela', 'gabriela.iana@gmail.com', '0712.623.881', 6, 1034);
insert into angajat
values(seq_cod_angajat.nextval, 'Tudor', 'Maria', 'maria.tudor@gmail.com', '0712.133.292', 10, null);
insert into angajat
values(seq_cod_angajat.nextval, 'Miron', 'Costin', 'costin.miron@gmail.com', '0771.229.345', 7, 1068);
insert into angajat
values(seq_cod_angajat.nextval, 'Banu', 'Codrin', 'codrin.banu@gmail.com', '0741.721.921', 9, null);
insert into angajat
values(seq_cod_angajat.nextval, 'Crivat', 'Gheorghe', 'gheo.crivat@gmail.com', null, 8, 1102);
insert into angajat
values(seq_cod_angajat.nextval, 'Stan', 'Elena', 'elena.stan@gmail.com', '0789.123.433', 10, null);
insert into angajat
values(seq_cod_angajat.nextval, 'Pitco', 'Natanael', 'nate.pitco@gmail.com', null, 7, 1136);

```

```

insert into angajat
values(seq_cod_angajat.nextval, 'Florea', 'Ana', 'ana.florea@gmail.com', null, 8, 1136);
insert into angajat
values(seq_cod_angajat.nextval, 'Nistor', 'Viviana', 'viv.nistor@gmail.com', '0712.385.763', 6, null);
insert into angajat
values(seq_cod_angajat.nextval, 'Leona', 'Antonia', 'anto.leona@gmail.com', '0711.332.454', 12, 1187);
insert into angajat
values(seq_cod_angajat.nextval, 'Nils', 'Corina', 'corina.nils@gmail.com', null, 8, 1187);
insert into angajat
values(seq_cod_angajat.nextval, 'Petre', 'Cosmin', 'cosmin.petre@gmail.com', '0718.223.344', 10, null);
insert into angajat
values(seq_cod_angajat.nextval, 'Popa', 'Maria', 'maria.popa@yahoo.com', '0792.233.211', 8, null);
insert into angajat
values(seq_cod_angajat.nextval, 'Dumitran', 'George', 'george.dumi@gmail.com', '0793.557.876', 9, null);

```

The screenshot shows the Oracle SQL Developer interface. A query window at the top contains the command:

```
select * from angajat;
```

Below it, a results window titled "Query Result" displays the following data:

COD_ANGAJAT	NUME	PRENUME	EMAIL	TELEFON	NR_ORE	ID_SUPERIOR	
1	1000	Popescu	Ion	ion.popescu@gmail.com	0741.311.314	8	(null)
2	1017	Georgesc	Vasile	vasile.georgesc@gmail.com	(null)	8	1000
3	1034	Matei	Andrei	andrei.matei@gmail.com	0799.532.678	6	1000
4	1051	Iana	Gabriela	gabriela.iana@gmail.com	0712.623.881	6	1034
5	1068	Tudor	Maria	maria.tudor@gmail.com	0712.133.292	10	(null)
6	1085	Miron	Costin	costin.miron@gmail.com	0771.229.345	7	1068
7	1102	Banu	Codrin	codrin.banu@gmail.com	0741.721.921	9	(null)
8	1119	Crivat	Gheorghe	gheo.crivat@gmail.com	(null)	8	1102
9	1136	Stan	Elena	elena.stan@gmail.com	0789.123.433	10	(null)
10	1153	Pitco	Natanael	nate.pitco@gmail.com	(null)	7	1136
11	1170	Florea	Ana	ana.florea@gmail.com	(null)	8	1136
12	1187	Nistor	Viviana	viv.nistor@gmail.com	0712.385.763	6	(null)
13	1204	Leona	Antonia	anto.leona@gmail.com	0711.332.454	12	1187
14	1221	Nils	Corina	corina.nils@gmail.com	(null)	8	1187
15	1238	Petre	Cosmin	cosmin.petre@gmail.com	0718.223.344	10	(null)
16	1255	Popa	Maria	maria.popa@yahoo.com	0792.233.211	8	(null)
17	1289	Dumitran	George	george.dumi@gmail.com	0793.557.876	9	(null)

Inserări în tabela CINEMATOGRAF:

```

insert into cinematograf
values(seq_cod_cinema.nextval, 'Cinema City', 105, 'Park Lake', '031.403.2700',
'cin.city.pklk@yahoo.com', '2 may 2016');

insert into cinematograf
values(seq_cod_cinema.nextval, 'Movieplex', 110, 'Plaza', '021.431.0000',
'movieplex.buc@yahoo.com', '23 june 2010');

insert into cinematograf
values(seq_cod_cinema.nextval, 'Cinema Arta Sibiu', 120, null, '0734.566.932',
'cin.artasibiu@yahoo.com', '27 december 2008');

insert into cinematograf

```

```

values(seq_cod_cinema.nextval, 'Cinema City', 125, 'Iulius Mall', '031.413.0400',
'cin.cityilm@yahoo.com', '16 april 2014');

insert into cinematograf
values(seq_cod_cinema.nextval, 'Cinema Victoria', 130, null, '0232.268.012',
'cin.victoria.is@yahoo.com', '4 july 2015');

insert into cinematograf
values(seq_cod_cinema.nextval, 'Cinema City', 135, null, null, 'cin.city.ct@yahoo.com',
'17 october 2018');

insert into cinematograf
values(seq_cod_cinema.nextval, 'Cinema Victoria', 140, null, '0744.338.939',
'c.victoria.cjn@yahoo.com', '30 august 2009');

insert into cinematograf
values(seq_cod_cinema.nextval, 'Cinema City', 145, null, '0362.403.030',
'cin.city.bm@yahoo.com', '7 may 2016');

insert into cinematograf
values(seq_cod_cinema.nextval, 'Cinema City', 105, 'Park Lake', '031.403.2700', 'cin.city.pklk@yahoo.com', '12 may 2016');
insert into cinematograf
values(seq_cod_cinema.nextval, 'Movieplex', 110, 'Plaza', '021.431.0000', 'movieplex.buc@yahoo.com', '23 june 2010');
insert into cinematograf
values(seq_cod_cinema.nextval, 'Cinema Arta Sibiu', 120, null, '0734.566.932', 'cin.artas.sibiu@yahoo.com', '27 december 2008');
insert into cinematograf
values(seq_cod_cinema.nextval, 'Cinema City', 125, 'Iulius Mall', '031.413.0400', 'cin.cityilm@yahoo.com', '16 april 2014');
insert into cinematograf
values(seq_cod_cinema.nextval, 'Cinema Victoria', 130, null, '0232.268.012', 'cin.victoria.is@yahoo.com', '4 july 2015');
insert into cinematograf
values(seq_cod_cinema.nextval, 'Cinema City', 135, null, null, 'cin.city.ct@yahoo.com', '17 october 2018');
insert into cinematograf
values(seq_cod_cinema.nextval, 'Cinema Victoria', 140, null, '0744.338.939', 'c.victoria.cjn@yahoo.com', '30 august 2009');
insert into cinematograf
values(seq_cod_cinema.nextval, 'Cinema City', 145, null, '0362.403.030', 'cin.city.bm@yahoo.com', '7 may 2016');

```

The screenshot shows the MySQL Workbench interface with a query editor and a results viewer. The query editor contains the SQL command:

```
select * from cinematograf;
```

The results viewer shows the data from the 'cinematograf' table:

	COD_CINEMA	NUME_CINEMA	ID_LOCATIE	COMPLEX	TELEFON	EMAIL	DATA_DESCH
1	120	Cinema City	105 Park Lake	031.403.2700	cin.city.pklk@yahoo.com	02-MAY-16	
2	140	Movieplex	110 Plaza	021.431.0000	movieplex.buc@yahoo.com	23-JUN-10	
3	160	Cinema Arta Sibiu	120 (null)	0734.566.932	cin.arta.sibiu@yahoo.com	27-DEC-08	
4	180	Cinema City	125 Iulius Mall	031.413.0400	cin.city.ilm@yahoo.com	16-APR-14	
5	200	Cinema Victoria	130 (null)	0232.268.012	cin.victoria.is@yahoo.com	04-JUL-15	
6	220	Cinema City	135 (null)	(null)	cin.city.ct@yahoo.com	17-OCT-18	
7	240	Cinema Victoria	140 (null)	0744.338.939	c.victoria.cjn@yahoo.com	30-AUG-09	
8	260	Cinema City	145 (null)	0362.403.030	cin.city.bm@yahoo.com	07-MAY-16	

Inserări în tabela **LUCREAZA**:

```
insert into lucreaza
values(120, 1000, 'MAN', '3 june 2018', null, 6023.5);
```

```
insert into lucreaza
values(120, 1017, 'AG_C', '3 may 2017', '13 june 2019', null);
```

```
insert into lucreaza
values(120, 1034, 'CAS', '27 january 2020', null, 4672.5);
```

```
insert into lucreaza
values(120, 1051, 'CAS', '12 april 2021', null, 3972.5);
```

```
insert into lucreaza
values(140, 1068, 'MAN', '5 september 2017', null, 5998.8);
```

```
insert into lucreaza
values(140, 1085, 'PLAS', '17 july 2021', '19 november 2019', null);
```

```
insert into lucreaza
values(180, 1102, 'MAN', '21 december 2018', null, 5772.6);
```

```
insert into lucreaza
values(180, 1119, 'ING_S', '13 february 2019', null, 4129.7);
```

```

insert into lucreaza
values(240, 1136, 'MAN', '9 july 2020', null, 5972.5);

insert into lucreaza
values(240, 1153, 'ING_T', '12 april 2022', null, 3472.5);

insert into lucreaza
values(240, 1170, 'BAR_C', '24 october 2021', null, 2121.3);

insert into lucreaza
values(160, 1187, 'MAN', '1 march 2019', null, 6100.5);

insert into lucreaza
values(160, 1204, 'BAR', '7 november 2020', '12 may 2022', null);

insert into lucreaza
values(160, 1221, 'ING_S', '26 june 2021', null, 5972.5);

insert into lucreaza
values(200, 1238, 'MAN', '2 may 2018', null, 6191.0);

insert into lucreaza
values(220, 1255, 'MAN', '15 may 2017', null, 6132.1);

insert into lucreaza
values(260, 1289, 'MAN', '19 november 2018', null, 5789.1);

```

```

insert into lucreaza
values(seq_cod_lucreaza.nextval, 120, 1000, 'MAN', 6023.5, '3 june 2018', null);
insert into lucreaza
values(seq_cod_lucreaza.nextval, 120, 1017, 'AG_C', 1500.0, '3 may 2017', '13 june 2019');
insert into lucreaza
values(seq_cod_lucreaza.nextval, 120, 1034, 'CAS', 4672.5, '27 january 2020', null);
insert into lucreaza
values(seq_cod_lucreaza.nextval, 120, 1051, 'CAS', 3972.5, '12 april 2021', null);
insert into lucreaza
values(seq_cod_lucreaza.nextval, 140, 1068, 'MAN', 5998.8, '5 september 2017', null);
insert into lucreaza
values(seq_cod_lucreaza.nextval, 140, 1085, 'PLAS', 1300.5, '17 july 2021', '19 november 2019');
insert into lucreaza
values(seq_cod_lucreaza.nextval, 180, 1102, 'MAN', 5772.6, '21 december 2018', null);
insert into lucreaza
values(seq_cod_lucreaza.nextval, 180, 1119, 'ING_S', 4129.7, '13 february 2019', null);
insert into lucreaza
values(seq_cod_lucreaza.nextval, 240, 1136, 'MAN', 5972.5, '9 july 2020', null);
insert into lucreaza
values(seq_cod_lucreaza.nextval, 240, 1153, 'ING_T', 3472.5, '12 april 2022', null);

```

```

insert into lucreaza
values(seq_cod_lucreaza.nextval, 240, 1170, 'BAR_C', 2121.3, '24 october 2021', null);
insert into lucreaza
values(seq_cod_lucreaza.nextval, 160, 1187, 'MAN', 6100.5, '1 march 2019', null);
insert into lucreaza
values(seq_cod_lucreaza.nextval, 160, 1204, 'BAR', 2671.1, '7 november 2020', '12 may 2022');
insert into lucreaza
values(seq_cod_lucreaza.nextval, 160, 1221, 'ING_S', 5972.5, '26 june 2021', null);
insert into lucreaza
values(seq_cod_lucreaza.nextval, 200, 1238, 'MAN', 6191.0, '2 may 2018', null);
insert into lucreaza
values(seq_cod_lucreaza.nextval, 220, 1255, 'MAN', 6132.1, '15 may 2017', null);
insert into lucreaza
values(seq_cod_lucreaza.nextval, 260, 1289, 'MAN', 5789.1, '19 november 2018', null);

```

```

| select * from lucreaza;
```

Script Output | Query Result | All Rows Fetched: 17 in 0.012 seconds

	COD_LUCRU	COD_CINEMA	COD_ANGAJAT	COD_JOB	SALARIU	DATA_ANG	DATA_DEMISIE
1	100	120	1000	MAN	6023.5	03-JUN-18	(null)
2	130	120	1017	AG_C	1500	03-MAY-17	13-JUN-19
3	160	120	1034	CAS	4672.5	27-JAN-20	(null)
4	190	120	1051	CAS	3972.5	12-APR-21	(null)
5	220	140	1068	MAN	5998.8	05-SEP-17	(null)
6	250	140	1085	PLAS	1300.5	17-JUL-21	19-NOV-19
7	280	180	1102	MAN	5772.6	21-DEC-18	(null)
8	310	180	1119	ING_S	4129.7	13-FEB-19	(null)
9	340	240	1136	MAN	5972.5	09-JUL-20	(null)
10	370	240	1153	ING_T	3472.5	12-APR-22	(null)
11	400	240	1170	BAR_C	2121.3	24-OCT-21	(null)
12	430	160	1187	MAN	6100.5	01-MAR-19	(null)
13	460	160	1204	BAR	2671.1	07-NOV-20	12-MAY-22
14	490	160	1221	ING_S	5972.5	26-JUN-21	(null)
15	520	200	1238	MAN	6191.0	02-MAY-18	(null)
16	550	220	1255	MAN	6132.1	15-MAY-17	(null)
17	580	260	1289	MAN	5789.1	19-NOV-18	(null)

Inserări în tabela SALA:

```

insert into sala
values(seq_nr_sala.nextval, 120, 'VIP');
```

```

insert into sala
values(seq_nr_sala.nextval, 120, 'Multiplex');
```

```

insert into sala
values(seq_nr_sala.nextval, 120, 'Megaplex');
```

```

insert into sala
values(seq_nr_sala.nextval, 140, 'Multiplex');
```

```

insert into sala
values(seq_nr_sala.nextval, 160, 'Complex');
```

```
insert into sala
values(seq_nr_sala.nextval, 160, 'VIP');

insert into sala
values(seq_nr_sala.nextval, 180, 'Megaplex');

insert into sala
values(seq_nr_sala.nextval, 200, 'Multiplex');

insert into sala
values(seq_nr_sala.nextval, 240, 'Complex');

insert into sala
values(seq_nr_sala.nextval, 220, 'VIP');

insert into sala
values(seq_nr_sala.nextval, 220, 'VIP');

insert into sala
values(seq_nr_sala.nextval, 200, 'Megaplex');

insert into sala
values(seq_nr_sala.nextval, 260, 'Multiplex');

insert into sala
values(seq_nr_sala.nextval, 260, 'Complex');

insert into sala
values(seq_nr_sala.nextval, 140, 'Complex');

insert into sala
values(seq_nr_sala.nextval, 200, 'Complex');

insert into sala
values(seq_nr_sala.nextval, 120, 'Megaplex');

insert into sala
values(seq_nr_sala.nextval, 120, 'VIP');

insert into sala
values(seq_nr_sala.nextval, 260, 'VIP');

insert into sala
values(seq_nr_sala.nextval, 220, 'Multiplex');
```

```

insert into sala
values(seq_nr_sala.nextval, 120, 'VIP');
insert into sala
values(seq_nr_sala.nextval, 120, 'Multiplex');
insert into sala
values(seq_nr_sala.nextval, 120, 'Megaplex');
insert into sala
values(seq_nr_sala.nextval, 140, 'Multiplex');
insert into sala
values(seq_nr_sala.nextval, 160, 'Complex');
insert into sala
values(seq_nr_sala.nextval, 160, 'VIP');
insert into sala
values(seq_nr_sala.nextval, 180, 'Megaplex');
insert into sala
values(seq_nr_sala.nextval, 200, 'Multiplex');
insert into sala
values(seq_nr_sala.nextval, 240, 'Complex');
insert into sala
values(seq_nr_sala.nextval, 220, 'VIP');
insert into sala
values(seq_nr_sala.nextval, 220, 'VIP');

insert into sala
values(seq_nr_sala.nextval, 200, 'Megaplex');
insert into sala
values(seq_nr_sala.nextval, 260, 'Multiplex');
insert into sala
values(seq_nr_sala.nextval, 260, 'Complex');
insert into sala
values(seq_nr_sala.nextval, 140, 'Complex');
insert into sala
values(seq_nr_sala.nextval, 200, 'Complex');
insert into sala
values(seq_nr_sala.nextval, 120, 'Megaplex');
insert into sala
values(seq_nr_sala.nextval, 120, 'VIP');
insert into sala
values(seq_nr_sala.nextval, 260, 'VIP');
insert into sala
values(seq_nr_sala.nextval, 220, 'Multiplex');

```

select * from sala;

Script Output | Query Result | All Rows Fetched: 20 in 0.014 seconds

	NR_SALA	COD_CINEMA	TIP
1	1	120	VIP
2	4	120	Multiplex
3	7	120	Megaplex
4	10	140	Multiplex
5	13	160	Complex
6	16	160	VIP
7	19	180	Megaplex
8	22	200	Multiplex
9	25	240	Complex
10	28	220	VIP
11	31	220	VIP
12	34	200	Megaplex
13	37	260	Multiplex

14	40	260 Complex
15	4	140 Complex
16	7	200 Complex
17	10	120 Megaplex
18	13	120 VIP
19	16	260 VIP
20	19	220 Multiplex

Inserări în tabela FILM:

```
insert into film
```

```
values(seq_cod_film.nextval, 'Uncharted', 'Actiune', 116, 6.4, 2022);
```

```
insert into film
```

```
values(seq_cod_film.nextval, 'Spiderman: No Way Home', 'Actiune', 148, 8.4, 2021);
```

```
insert into film
```

```
values(seq_cod_film.nextval, 'Free Guy', 'Actiune, Comedie', 115, 7.1, 2021);
```

```
insert into film
```

```
values(seq_cod_film.nextval, 'Little Women', 'Romance, Drama', 135, 7.8, 2019);
```

```
insert into film
```

```
values(seq_cod_film.nextval, 'It', 'Horror', 135, 7.4, 2017);
```

```
insert into film
```

```
values(seq_cod_film.nextval, 'Encanto', 'Animatie', 109, 7.8, 2021);
```

```
insert into film
```

```
values(seq_cod_film.nextval, 'Star Wars: Episode III', 'S.F.', 140, 7.6, 2005);
```

```
insert into film
```

```
values(seq_cod_film.nextval, 'Black Widow', 'Actiune', 134, 6.4, 2021);
```

```
insert into film
```

```
values(seq_cod_film.nextval, 'Titanic', 'Romance, Drama', 194, 7.9, 1997);
```

```
insert into film
```

```
values(seq_cod_film.nextval, 'The Batman', 'Actiune, Crima', 176, 8.0, 2022);
```

```
insert into film
```

```
values(seq_cod_film.nextval, 'Red Notice', 'Actiune, Comedie', 118, 6.3, 2021);
```

```
insert into film
```

```
values(seq_cod_film.nextval, 'Venom', 'Actiune, Horror', 112, 7.9, 2018);
```

```
insert into film
values(seq_cod_film.nextval, 'Crimson Peak', 'Horror, Mister', 119, 7.0, 2015);
```

```
insert into film
values(seq_cod_film.nextval, 'Yes Day', 'Comedie, Familie', 146, 5.7, 2021);
```

```
insert into film
values(seq_cod_film.nextval, 'Zootopia', 'Animatie', 108, 8.5, 2016);
```

```
insert into film
values(seq_cod_film.nextval, 'Doctor Strange', 'Actiune, S.F.', 116, 7.5, 2016);
```

```
insert into film
values(seq_cod_film.nextval, 'It chapter 2', 'Horror', 169, 6.5, 2019);
```

```
insert into film
values(seq_cod_film.nextval, 'Death on the Nile', 'Crima, Mister', 127, 6.9, 2022);
```

```
insert into film
values(seq_cod_film.nextval, 'Sherlock Holmes', 'Crima, Mister', 128, 8.6, 2009);
```

```
insert into film
values(seq_cod_film.nextval, 'Divergent', 'Romance, S.F.', 140, 8.1, 2014);
```

```
insert into film
values(seq_cod_film.nextval, 'Uncharted', 'Actiune', 116, 6.4, 2022);
insert into film
values(seq_cod_film.nextval, 'Spiderman: No Way Home', 'Actiune', 148, 8.4, 2021);
insert into film
values(seq_cod_film.nextval, 'Free Guy', 'Actiune, Comedie', 115, 7.1, 2021);
insert into film
values(seq_cod_film.nextval, 'Little Women', 'Romance, Drama', 135, 7.8, 2019);
insert into film
values(seq_cod_film.nextval, 'It', 'Horror', 135, 7.4, 2017);
insert into film
values(seq_cod_film.nextval, 'Encanto', 'Animatie', 109, 7.8, 2021);
insert into film
values(seq_cod_film.nextval, 'Star Wars: Episode III', 'S.F.', 140, 7.6, 2005);
insert into film
values(seq_cod_film.nextval, 'Black Widow', 'Actiune', 134, 6.4, 2021);
insert into film
values(seq_cod_film.nextval, 'Titanic', 'Romance, Drama', 194, 7.9, 1997);
insert into film
values(seq_cod_film.nextval, 'The Batman', 'Actiune, Crima', 176, 8.0, 2022);
insert into film
values(seq_cod_film.nextval, 'Red Notice', 'Actiune, Comedie', 118, 6.3, 2021);
```

```

insert into film
values(seq_cod_film.nextval, 'Venom', 'Actiune, Horror', 112, 7.9, 2018);
insert into film
values(seq_cod_film.nextval, 'Crimson Peak', 'Horror, Mister', 119, 7.0, 2015);
insert into film
values(seq_cod_film.nextval, 'Yes Day', 'Comedie, Familie', 146, 5.7, 2021);
insert into film
values(seq_cod_film.nextval, 'Zootopia', 'Animatie', 108, 8.5, 2016);
insert into film
values(seq_cod_film.nextval, 'Doctor Strange', 'Actiune, S.F.', 116, 7.5, 2016);
insert into film
values(seq_cod_film.nextval, 'It chapter 2', 'Horror', 169, 6.5, 2019);
insert into film
values(seq_cod_film.nextval, 'Death on the Nile', 'Crima, Mister', 127, 6.9, 2022);
insert into film
values(seq_cod_film.nextval, 'Sherlock Holmes', 'Crima, Mister', 128, 8.6, 2009);
insert into film
values(seq_cod_film.nextval, 'Divergent', 'Romance, S.F.', 140, 8.1, 2014);

```

select * from film;

COD_FILM	NUME_FILM	GEN	DURATA	RATING	AN_APARITIE
1	10 Uncharted	Actiune	116	6.4	2022
2	110 Spiderman: No Way Home	Actiune	148	8.4	2021
3	210 Free Guy	Actiune, Comedie	115	7.1	2021
4	310 Little Women	Romance, Drama	135	7.8	2019
5	410 It	Horror	135	7.4	2017
6	510 Encanto	Animatie	109	7.8	2021
7	610 Star Wars: Episode III S.F.		140	7.6	2005
8	710 Black Widow	Actiune	134	6.4	2021
9	810 Titanic	Romance, Drama	194	7.9	1997
10	910 The Batman	Actiune, Crima	176	8	2022
11	1010 Red Notice	Actiune, Comedie	118	6.3	2021
12	1110 Venom	Actiune, Horror	112	7.9	2018
13	1210 Crimson Peak	Horror, Mister	119	7	2015
14	1310 Yes Day	Comedie, Familie	146	5.7	2021
15	1410 Zootopia	Animatie	108	8.5	2016
16	1510 Doctor Strange	Actiune, S.F.	116	7.5	2016
17	1610 It chapter 2	Horror	169	6.5	2019
18	1710 Death on the Nile	Crima, Mister	127	6.9	2022
19	1810 Sherlock Holmes	Crima, Mister	128	8.6	2009
20	1910 Divergent	Romance, S.F.	140	8.1	2014

Inserări în tabela RULEAZA:

```

insert into ruleaza
values(seq_cod_ruleaza.nextval, 120, 10, '2 march 2022', '13 may 2022');

```

```

insert into ruleaza

```

```
values(seq_cod_ruleaza.nextval, 120, 1210, '21 june 2018', '13 january 2019');

insert into ruleaza
values(seq_cod_ruleaza.nextval, 140, 1110, '17 july 2020', '23 november 2020');

insert into ruleaza
values(seq_cod_ruleaza.nextval, 180, 1110, '17 july 2020', '23 november 2020');

insert into ruleaza
values(seq_cod_ruleaza.nextval, 200, 810, '15 april 2018', '19 may 2019');

insert into ruleaza
values(seq_cod_ruleaza.nextval, 200, 410, '5 september 2017', '16 december 2017');

insert into ruleaza
values(seq_cod_ruleaza.nextval, 200, 1610, '15 november 2019', '8 february 2020');

insert into ruleaza
values(seq_cod_ruleaza.nextval, 160, 1610, '15 november 2019', '8 february 2020');

insert into ruleaza
values(seq_cod_ruleaza.nextval, 140, 710, '28 may 2021', '13 october 2021');

insert into ruleaza
values(seq_cod_ruleaza.nextval, 180, 1210, '23 december 2015', '27 may 2016');

insert into ruleaza
values(seq_cod_ruleaza.nextval, 180, 510, '1 january 2022', '23 april 2022');

insert into ruleaza
values(seq_cod_ruleaza.nextval, 220, 910, '15 march 2022', '23 july 2022');

insert into ruleaza
values(seq_cod_ruleaza.nextval, 220, 1710, '15 april 2022', '3 september 2022');

insert into ruleaza
values(seq_cod_ruleaza.nextval, 200, 610, '5 june 2015', '18 august 2015');

insert into ruleaza
values(seq_cod_ruleaza.nextval, 240, 610, '6 june 2015', '18 september 2015');

insert into ruleaza
values(seq_cod_ruleaza.nextval, 160, 1310, '6 april 2021', '14 november 2021');--
```



```
insert into ruleaza
values(seq_cod_ruleaza.nextval, 120, 1610, '15 november 2019', '8 february 2020');
```

```
insert into ruleaza
values(seq_cod_ruleaza.nextval, 200, 710, '28 may 2021', '13 october 2021');

insert into ruleaza
values(seq_cod_ruleaza.nextval, 240, 1210, '23 december 2015', '27 may 2016');

insert into ruleaza
values(seq_cod_ruleaza.nextval, 240, 510, '1 january 2022', '23 april 2022');

insert into ruleaza
values(seq_cod_ruleaza.nextval, 260, 1510, '19 may 2017', '21 december 2017');

insert into ruleaza
values(seq_cod_ruleaza.nextval, 260, 310, '15 august 2019', '5 january 2020');

insert into ruleaza
values(seq_cod_ruleaza.nextval, 120, 310, '15 august 2019', '5 january 2020');

insert into ruleaza
values(seq_cod_ruleaza.nextval, 160, 310, '16 august 2019', '5 january 2020');

insert into ruleaza
values(seq_cod_ruleaza.nextval, 220, 1010, '15 september 2021', '14 february 2022');

insert into ruleaza
values(seq_cod_ruleaza.nextval, 240, 1010, '17 september 2021', '14 february 2022');

insert into ruleaza
values(seq_cod_ruleaza.nextval, 120, 1010, '16 september 2021', '11 february 2022');

insert into ruleaza
values(seq_cod_ruleaza.nextval, 140, 1310, '6 april 2021', '14 november 2021');

insert into ruleaza
values(seq_cod_ruleaza.nextval, 140, 110, '16 december 2021', '14 february 2022');

insert into ruleaza
values(seq_cod_ruleaza.nextval, 160, 210, '6 july 2021', '19 october 2021');
```

```

insert into ruleaza
values(seq_cod_ruleaza.nextval, 120, 10, '2 march 2022', '13 may 2022');
insert into ruleaza
values(seq_cod_ruleaza.nextval, 120, 1210, '21 june 2018', '13 january 2019');
insert into ruleaza
values(seq_cod_ruleaza.nextval, 140, 1110, '17 july 2020', '23 november 2020');
insert into ruleaza
values(seq_cod_ruleaza.nextval, 180, 1110, '17 july 2020', '23 november 2020');
insert into ruleaza
values(seq_cod_ruleaza.nextval, 200, 810, '15 april 2018', '19 may 2019');
insert into ruleaza
values(seq_cod_ruleaza.nextval, 200, 410, '5 september 2017', '16 december 2017');
insert into ruleaza
values(seq_cod_ruleaza.nextval, 200, 1610, '15 november 2019', '8 february 2020');
insert into ruleaza
values(seq_cod_ruleaza.nextval, 160, 1610, '15 november 2019', '8 february 2020');
insert into ruleaza
values(seq_cod_ruleaza.nextval, 140, 710, '28 may 2021', '13 october 2021');

insert into ruleaza
values(seq_cod_ruleaza.nextval, 180, 1210, '23 december 2015', '27 may 2016');
insert into ruleaza
values(seq_cod_ruleaza.nextval, 180, 510, '1 january 2022', '23 april 2022');
insert into ruleaza
values(seq_cod_ruleaza.nextval, 220, 910, '15 march 2022', '23 july 2022');
insert into ruleaza
values(seq_cod_ruleaza.nextval, 220, 1710, '15 april 2022', '3 september 2022');
insert into ruleaza
values(seq_cod_ruleaza.nextval, 200, 610, '5 june 2015', '18 august 2015');
insert into ruleaza
values(seq_cod_ruleaza.nextval, 240, 610, '6 june 2015', '18 september 2015');
insert into ruleaza
values(seq_cod_ruleaza.nextval, 160, 1310, '6 april 2021', '14 november 2021');
insert into ruleaza
values(seq_cod_ruleaza.nextval, 120, 1610, '15 november 2019', '8 february 2020');
insert into ruleaza
values(seq_cod_ruleaza.nextval, 200, 710, '28 may 2021', '13 october 2021');

insert into ruleaza
values(seq_cod_ruleaza.nextval, 260, 1510, '19 may 2017', '21 december 2017');
insert into ruleaza
values(seq_cod_ruleaza.nextval, 260, 310, '15 august 2019', '5 january 2020');
insert into ruleaza
values(seq_cod_ruleaza.nextval, 120, 310, '15 august 2019', '5 january 2020');
insert into ruleaza
values(seq_cod_ruleaza.nextval, 160, 310, '16 august 2019', '5 january 2020');
insert into ruleaza
values(seq_cod_ruleaza.nextval, 220, 1010, '15 september 2021', '14 february 2022');
insert into ruleaza
values(seq_cod_ruleaza.nextval, 240, 1010, '17 september 2021', '14 february 2022');
insert into ruleaza
values(seq_cod_ruleaza.nextval, 120, 1010, '16 september 2021', '11 february 2022');
insert into ruleaza
values(seq_cod_ruleaza.nextval, 140, 1310, '6 april 2021', '14 november 2021');
insert into ruleaza
values(seq_cod_ruleaza.nextval, 140, 110, '16 december 2021', '14 february 2022');
insert into ruleaza
values(seq_cod_ruleaza.nextval, 160, 210, '6 july 2021', '19 october 2021');

```

| select * from ruleaza;

Script Output x Query Result x

SQL | All Rows Fetched: 30 in 0.002 seconds

	COD_RULARE	COD_CINEMA	COD_FILM	DATA_INCEPUT	DATA_FINAL
1	260	120	10 02-MAR-22	13-MAY-22	
2	300	120	1210 21-JUN-18	13-JAN-19	
3	340	140	1110 17-JUL-20	23-NOV-20	
4	380	180	1110 17-JUL-20	23-NOV-20	
5	420	200	810 15-APR-18	19-MAY-19	
6	460	200	410 05-SEP-17	16-DEC-17	
7	500	200	1610 15-NOV-19	08-FEB-20	
8	540	160	1610 15-NOV-19	08-FEB-20	
9	580	140	710 28-MAY-21	13-OCT-21	
10	620	180	1210 23-DEC-15	27-MAY-16	
11	660	180	510 01-JAN-22	23-APR-22	
12	700	220	910 15-MAR-22	23-JUL-22	
13	740	220	1710 15-APR-22	03-SEP-22	
14	780	200	610 05-JUN-15	18-AUG-15	
15	820	240	610 06-JUN-15	18-SEP-15	
16	860	160	1310 06-APR-21	14-NOV-21	
17	900	120	1610 15-NOV-19	08-FEB-20	
18	940	200	710 28-MAY-21	13-OCT-21	
19	980	240	1210 23-DEC-15	27-MAY-16	
20	1020	240	510 01-JAN-22	23-APR-22	
21	1060	260	1510 19-MAY-17	21-DEC-17	
22	1100	260	310 15-AUG-19	05-JAN-20	
23	1140	120	310 15-AUG-19	05-JAN-20	
24	1180	160	310 16-AUG-19	05-JAN-20	
25	1220	220	1010 15-SEP-21	14-FEB-22	
26	1260	240	1010 17-SEP-21	14-FEB-22	
27	1300	120	1010 16-SEP-21	11-FEB-22	
28	1340	140	1310 06-APR-21	14-NOV-21	
29	1380	140	110 16-DEC-21	14-FEB-22	
30	1420	160	210 06-JUL-21	19-OCT-21	

Inserări în tabela **DIFUZARE**:

```
insert into difuzare
values(seq_cod_difuzare.nextval, 10, 1, 120, '3 april 2022', 19);
```

```
insert into difuzare
values(seq_cod_difuzare.nextval, 1210, 7, 120, '9 july 2018', 12);
```

```
insert into difuzare
values(seq_cod_difuzare.nextval, 710, 10, 140, '5 august 2021', 15);
```

```
insert into difuzare
values(seq_cod_difuzare.nextval, 1510, 37, 260, '19 may 2017', 9);
```

```
insert into difuzare
values(seq_cod_difuzare.nextval, 1110, 4, 140, '3 august 2022', 10);
```

```
insert into difuzare
values(seq_cod_difuzare.nextval, 1010, 28, 220, '19 december 2018', 11);

insert into difuzare
values(seq_cod_difuzare.nextval, 610, 34, 200, '18 august 2015', 14);

insert into difuzare
values(seq_cod_difuzare.nextval, 410, 22, 200, '15 november 2017', 9);

insert into difuzare
values(seq_cod_difuzare.nextval, 1310, 16, 160, '9 july 2021', 19);

insert into difuzare
values(seq_cod_difuzare.nextval, 1610, 16, 160, '9 july 2018', 22);

insert into difuzare
values(seq_cod_difuzare.nextval, 1010, 25, 240, '18 september 2021', 11);

insert into difuzare
values(seq_cod_difuzare.nextval, 1210, 25, 240, '1 march 2016', 18);

insert into difuzare
values(seq_cod_difuzare.nextval, 910, 28, 220, '3 july 2022', 17);

insert into difuzare
values(seq_cod_difuzare.nextval, 1210, 4, 120, '7 october 2018', 11);

insert into difuzare
values(seq_cod_difuzare.nextval, 710, 7, 200, '27 june 2021', 16);

insert into difuzare
values(seq_cod_difuzare.nextval, 1510, 40, 260, '7 august 2017', 12);

insert into difuzare
values(seq_cod_difuzare.nextval, 1710, 31, 220, '1 september 2022', 13);

insert into difuzare
values(seq_cod_difuzare.nextval, 1310, 10, 140, '6 june 2021', 22);

insert into difuzare
values(seq_cod_difuzare.nextval, 510, 19, 180, '2 february 2022', 19);

insert into difuzare
values(seq_cod_difuzare.nextval, 1210, 19, 180, '2 february 2022', 12);
```

```

insert into difuzare
values(seq_cod_difuzare.nextval, 10, 1, 120, '3 april 2022', 19);
insert into difuzare
values(seq_cod_difuzare.nextval, 1210, 7, 120, '9 july 2018', 12);
insert into difuzare
values(seq_cod_difuzare.nextval, 710, 10, 140, '5 august 2021', 15);
insert into difuzare
values(seq_cod_difuzare.nextval, 1510, 37, 260, '19 may 2017', 9);
insert into difuzare
values(seq_cod_difuzare.nextval, 1110, 4, 140, '3 august 2022', 10);
insert into difuzare
values(seq_cod_difuzare.nextval, 1010, 28, 220, '19 december 2018', 11);
insert into difuzare
values(seq_cod_difuzare.nextval, 610, 34, 200, '18 august 2015', 14);
insert into difuzare
values(seq_cod_difuzare.nextval, 410, 22, 200, '15 november 2017', 9);
insert into difuzare
values(seq_cod_difuzare.nextval, 1310, 16, 160, '9 july 2021', 19);
insert into difuzare
values(seq_cod_difuzare.nextval, 1610, 16, 160, '9 july 2018', 22);
insert into difuzare
values(seq_cod_difuzare.nextval, 1010, 25, 240, '18 september 2021', 11);

insert into difuzare
values(seq_cod_difuzare.nextval, 1210, 25, 240, '1 march 2016', 18);
insert into difuzare
values(seq_cod_difuzare.nextval, 910, 28, 220, '3 july 2022', 17);
insert into difuzare
values(seq_cod_difuzare.nextval, 1210, 4, 120, '7 october 2018', 11);
insert into difuzare
values(seq_cod_difuzare.nextval, 710, 7, 200, '27 june 2021', 16);
insert into difuzare
values(seq_cod_difuzare.nextval, 1510, 40, 260, '7 august 2017', 12);
insert into difuzare
values(seq_cod_difuzare.nextval, 1710, 31, 220, '1 september 2022', 13);
insert into difuzare
values(seq_cod_difuzare.nextval, 1310, 10, 140, '6 june 2021', 22);
insert into difuzare
values(seq_cod_difuzare.nextval, 510, 19, 180, '2 february 2022', 19);
insert into difuzare
values(seq_cod_difuzare.nextval, 1210, 19, 180, '2 february 2022', 12);

```

	COD_DIFUZARE	COD_FILM	NR_SALA	COD_CINEMA	DATA_INC	ORA_INC
1	50	10	1	120	03-APR-22	19
2	71	1210	7	120	09-JUL-18	12
3	92	710	10	140	05-AUG-21	15
4	113	1510	37	260	19-MAY-17	9
5	155	1110	4	140	03-AUG-22	10
6	176	1010	28	220	19-DEC-18	11
7	197	610	34	200	18-AUG-15	14
8	218	410	22	200	15-NOV-17	9
9	239	1310	16	160	09-JUL-21	19
10	260	1610	16	160	09-JUL-18	22

11	281	1010	25	24018-SEP-21	11
12	323	1210	25	24001-MAR-16	18
13	344	910	28	22003-JUL-22	17
14	365	1210	4	12007-OCT-18	11
15	386	710	7	20027-JUN-21	16
16	407	1510	40	26007-AUG-17	12
17	428	1710	31	22001-SEP-22	13
18	449	1310	10	14006-JUN-21	22
19	470	510	19	18002-FEB-22	19
20	491	1210	19	18002-FEB-22	12

Inserări în tabela **BILET**:

```

insert into bilet
values(seq_cod_bilet.nextval, 17.1, 17, 3, 50);

insert into bilet
values(seq_cod_bilet.nextval, 22.5, 1, 1, 386);

insert into bilet
values(seq_cod_bilet.nextval, 18.6, 78, 10, 323);

insert into bilet
values(seq_cod_bilet.nextval, 36.9, 22, 5, 491);

insert into bilet
values(seq_cod_bilet.nextval, 0.0, 15, 2, 491);

insert into bilet
values(seq_cod_bilet.nextval, 24.3, 98, 14, 113);

insert into bilet
values(seq_cod_bilet.nextval, 22.5, 56, 4, 176);

insert into bilet
values(seq_cod_bilet.nextval, 40.1, 27, 3, 92);

insert into bilet
values(seq_cod_bilet.nextval, 10.4, 61, 7, 260);

insert into bilet
values(seq_cod_bilet.nextval, 12.2, 15, 3, 407);

```

```

insert into bilet
values(seq_cod_bilet.nextval, 17.1, 17, 3, 50);
insert into bilet
values(seq_cod_bilet.nextval, 22.5, 1, 1, 386);
insert into bilet
values(seq_cod_bilet.nextval, 18.6, 78, 10, 323);
insert into bilet
values(seq_cod_bilet.nextval, 36.9, 22, 5, 491);
insert into bilet
values(seq_cod_bilet.nextval, 0.0, 15, 2, 491);
insert into bilet
values(seq_cod_bilet.nextval, 24.3, 98, 14, 113);
insert into bilet
values(seq_cod_bilet.nextval, 22.5, 56, 4, 176);
insert into bilet
values(seq_cod_bilet.nextval, 40.1, 27, 3, 92);
insert into bilet
values(seq_cod_bilet.nextval, 10.4, 61, 7, 260);
insert into bilet
values(seq_cod_bilet.nextval, 12.2, 15, 3, 407);

```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a script editor containing the SQL code for inserting data into the 'bilet' table. In the bottom-left pane, there is a query result viewer showing the output of the executed query. The results are displayed in a table with columns: COD_BILET, PRET, NR_LOC, NR_RAND, and COD_DIFUZARE. The data consists of 10 rows of values.

COD_BILET	PRET	NR_LOC	NR_RAND	COD_DIFUZARE
1	11	17.1	17	3
2	16	22.5	1	1
3	21	18.6	78	10
4	26	36.9	22	5
5	31	0	15	2
6	36	24.3	98	14
7	41	22.5	56	4
8	46	12.2	15	3
9	51	10.4	61	7
10	56	40.1	27	3

Inserări în tabela ACTOR:

```

insert into actor
values(seq_id_actor.nextval, 'Zendaya', null, 'SUA', 22, 2009, 67898.9);

```

```

insert into actor
values(seq_id_actor.nextval, 'Tom', 'Holland', 'EN', 7, 2008, 55118.3);

```

```

insert into actor

```

```
values(seq_id_actor.nextval, 'Ryan', 'Reynolds', 'CA', 9, 1991, 82658.5);

insert into actor
values(seq_id_actor.nextval, 'Emma', 'Watson', 'EN', 24, 2001, 72628.4);

insert into actor
values(seq_id_actor.nextval, 'Finn', 'Wolfhard', 'CA', 3, 2017, 32267.9);

insert into actor
values(seq_id_actor.nextval, 'Gal', 'Gadot', 'IL', 3, 2004, 56118.3);

insert into actor
values(seq_id_actor.nextval, 'Robert', 'Downey Jr', 'SUA', 29, 1970, 89658.5);

insert into actor
values(seq_id_actor.nextval, 'Robert', 'Pattinson', 'EN', 30, 2005, 76228.4);

insert into actor
values(seq_id_actor.nextval, 'Zendaya', null, 'SUA', 22, 2009, 67898.9);
insert into actor
values(seq_id_actor.nextval, 'Tom', 'Holland', 'EN', 7, 2008, 55118.3);
insert into actor
values(seq_id_actor.nextval, 'Ryan', 'Reynolds', 'CA', 9, 1991, 82658.5);
insert into actor
values(seq_id_actor.nextval, 'Emma', 'Watson', 'EN', 24, 2001, 72628.4);
insert into actor
values(seq_id_actor.nextval, 'Finn', 'Wolfhard', 'CA', 3, 2017, 32267.9);
insert into actor
values(seq_id_actor.nextval, 'Gal', 'Gadot', 'IL', 3, 2004, 56118.3);
insert into actor
values(seq_id_actor.nextval, 'Robert', 'Downey Jr', 'SUA', 29, 1970, 89658.5);
insert into actor
values(seq_id_actor.nextval, 'Robert', 'Pattinson', 'EN', 30, 2005, 76228.4);
```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor window containing the SQL command:

```
select * from actor;
```

In the bottom-right pane, there is a "Query Result" window titled "Query Result X". It displays the results of the executed query. The results are presented in a table with the following columns:

ID_ACTOR	PRENUME	NUME	TARA	NR_PREMII	AN_DEBUT	SALARIU
1	10	Tom	Holland	EN	7	2008 55118.3
2	40	Ryan	Reynolds	CA	9	1991 82658.5
3	55	Emma	Watson	EN	24	2001 72628.4
4	70	Finn	Wolfhard	CA	3	2017 32267.9
5	85	Gal	Gadot	IL	3	2004 56118.3
6	100	Robert	Downey Jr	SUA	29	1970 89658.5
7	115	Robert	Pattinson	EN	30	2005 76228.4
8	25	(null)	Zendaya	SUA	22	2009 67898.9

The status bar at the bottom of the "Query Result" window indicates "All Rows Fetched: 8 in 0.006 seconds".

Inserări în tabela JOACA:

```
insert into joaca
values(seq_cod_joaca.nextval, 10, 10, 'principal');

insert into joaca
values(seq_cod_joaca.nextval, 110, 10, 'principal');

insert into joaca
values(seq_cod_joaca.nextval, 410, 70, 'principal');

insert into joaca
values(seq_cod_joaca.nextval, 1610, 70, 'secundar');

insert into joaca
values(seq_cod_joaca.nextval, 210, 40, 'principal');

insert into joaca
values(seq_cod_joaca.nextval, 1010, 40, 'principal');

insert into joaca
values(seq_cod_joaca.nextval, 310, 55, 'principal');

insert into joaca
```

```
values(seq_cod_joaca.nextval, 110, 25, 'principal');

insert into joaca
values(seq_cod_joaca.nextval, 910, 115, 'principal');

insert into joaca
values(seq_cod_joaca.nextval, 1710, 85, 'principal');

insert into joaca
values(seq_cod_joaca.nextval, 1810, 100, 'principal');

insert into joaca
values(seq_cod_joaca.nextval, 10, 10, 'principal');
insert into joaca
values(seq_cod_joaca.nextval, 110, 10, 'principal');
insert into joaca
values(seq_cod_joaca.nextval, 410, 70, 'principal');
insert into joaca
values(seq_cod_joaca.nextval, 1610, 70, 'secundar');
insert into joaca
values(seq_cod_joaca.nextval, 210, 40, 'principal');
insert into joaca
values(seq_cod_joaca.nextval, 1010, 40, 'principal');
insert into joaca
values(seq_cod_joaca.nextval, 310, 55, 'principal');
insert into joaca
values(seq_cod_joaca.nextval, 110, 25, 'principal');
insert into joaca
values(seq_cod_joaca.nextval, 910, 115, 'principal');
```

```
insert into joaca
values(seq_cod_joaca.nextval, 1710, 85, 'principal');
insert into joaca
values(seq_cod_joaca.nextval, 1810, 100, 'principal');
```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor with the following SQL query:

```
select * from joaca;
```

In the bottom-right pane, the results of the query are displayed in a grid format. The grid has four columns with the following headers:

COD_JOACA	COD_FILM	ID_ACTOR	ROL
1	30	10	10 principal
2	50	110	10 principal
3	70	410	70 principal
4	90	1610	70 secundar
5	110	210	40 principal
6	130	1010	40 principal
7	150	310	55 principal
8	170	110	25 principal
9	190	910	115 principal
10	210	1710	85 principal
11	230	1810	100 principal

The status bar at the bottom of the interface indicates "All Rows Fetched: 11 in 0.008 seconds".

Inserări în tabela **REGIZOR**:

```
insert into regizor
values(seq_id_regizor.nextval, 'Jon', 'Watts', 'SUA', 0);

insert into regizor
values(seq_id_regizor.nextval, 'Shawn', 'Levy', 'CA', 0);

insert into regizor
values(seq_id_regizor.nextval, 'Matt', 'Reeves', 'SUA', 1);

insert into regizor
values(seq_id_regizor.nextval, 'Greta', 'Gerwig', 'SUA', 6);

insert into regizor
values(seq_id_regizor.nextval, 'George', 'Lucas', 'SUA', 7);

insert into regizor
values(seq_id_regizor.nextval, 'Ruben', 'Fleischer', 'SUA', 0);

insert into regizor
values(seq_id_regizor.nextval, 'Andy', 'Muschietti', 'AR', 0);

insert into regizor
values(seq_id_regizor.nextval, 'Kenneth', 'Branagh', 'EN', 22);

insert into regizor
```

```

values(seq_id_regizor.nextval, 'Guy', 'Ritchie', 'EN', 5);

insert into regizor
values(seq_id_regizor.nextval, 'Andy', 'Serkis', 'EN', 8);

insert into regizor
values(seq_id_regizor.nextval, 'Jon', 'Watts', 'SUA', 0);
insert into regizor
values(seq_id_regizor.nextval, 'Shawn', 'Levy', 'CA', 0);
insert into regizor
values(seq_id_regizor.nextval, 'Matt', 'Reeves', 'SUA', 1);
insert into regizor
values(seq_id_regizor.nextval, 'Greta', 'Gerwig', 'SUA', 6);
insert into regizor
values(seq_id_regizor.nextval, 'George', 'Lucas', 'SUA', 7);
insert into regizor
values(seq_id_regizor.nextval, 'Ruben', 'Fleischer', 'SUA', 0);
insert into regizor
values(seq_id_regizor.nextval, 'Andy', 'Muschietti', 'AR', 0);
insert into regizor
values(seq_id_regizor.nextval, 'Kenneth', 'Branagh', 'EN', 22);
insert into regizor
values(seq_id_regizor.nextval, 'Guy', 'Ritchie', 'EN', 5);
insert into regizor
values(seq_id_regizor.nextval, 'Andy', 'Serkis', 'EN', 8);

```

select * from regizor;

The screenshot shows the Oracle SQL Developer interface. A query window is open with the SQL command: "select * from regizor;". Below it, a "Query Result" window displays the data from the "regizor" table. The table has five columns: ID_REGIZOR, PRENUME, NUME, TARA, and NR_PREMII. The data consists of 10 rows, each representing a director with their name, nationality, and number of awards.

ID_REGIZOR	PRENUME	NUME	TARA	NR_PREMII	
1	30	Jon	Watts	SUA	0
2	50	Shawn	Levy	CA	0
3	70	Matt	Reeves	SUA	1
4	90	Greta	Gerwig	SUA	6
5	110	George	Lucas	SUA	7
6	130	Ruben	Fleischer	SUA	0
7	150	Andy	Muschietti	AR	0
8	170	Kenneth	Branagh	EN	22
9	190	Guy	Ritchie	EN	5
10	210	Andy	Serkis	EN	8

Inserări în tabela INDRUMA:

```

insert into indruma
values(1110, 210, 2017);

```

```
insert into indruma
values(1110, 130, 2017);

insert into indruma
values(610, 110, 2003);

insert into indruma
values(110, 30, 2020);

insert into indruma
values(210, 50, 2020);

insert into indruma
values(910, 70, 2021);

insert into indruma
values(310, 90, 2017);

insert into indruma
values(10, 130, 2021);

insert into indruma
values(410, 150, 2015);

insert into indruma
values(1610, 150, 2017);

insert into indruma
values(1710, 170, 2020);

insert into indruma
values(1810, 190, 2008);
```

```
insert into indruma
values(1110, 210, 2017);
insert into indruma
values(1110, 130, 2017);
insert into indruma
values(610, 110, 2003);
insert into indruma
values(110, 30, 2020);
insert into indruma
values(210, 50, 2020);
insert into indruma
values(910, 70, 2021);
insert into indruma
values(310, 90, 2017);
```

```
insert into indruma
values(10, 130, 2021);
insert into indruma
values(410, 150, 2015);
insert into indruma
values(1610, 150, 2017);
insert into indruma
values(1710, 170, 2020);
insert into indruma
values(1810, 190, 2008);
```

```
select * from indruma;
```

	COD_FILM	ID_REGIZOR	AN_INCEPUT
1	1110	210	2017
2	1110	130	2017
3	610	110	2003
4	110	30	2020
5	210	50	2020
6	910	70	2021
7	310	90	2017
8	10	130	2021
9	410	150	2015
10	1610	150	2017
11	1710	170	2020
12	1810	190	2008

12. Cereri SQL

În acest subcapitol au fost formulate în limbaj natural și implementate 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:

- operație join pe cel puțin 4 tabele
- filtrare la nivel de linii
- subcereri sincronizate în care intervin cel puțin 3 tabele
- subcereri nesincronizate în care intervin cel puțin 3 tabele
- grupări de date, funcții grup, filtrare la nivel de grupuri
- ordonări
- utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a funcțiilor NVL și DECODE, a cel puțin unei expresii CASE
- utilizarea a cel puțin 1 bloc de cerere (clauza WITH)

- I. Afişați codul cinematografului, codul filmului, numele filmului, numărul sălii în care va difuza filmul, numele și prenumele (concatenate) actorilor care joacă în acest film și aprecierea filmului. Apreciera este obținută prin următarea regulă: dacă filmul are un rating de nota 8, atunci este „Foarte apreciat”, dacă are un rating de 6.3 este „Nu foarte apreciat”, altfel este „Apreciat”. Rezultatele trebuie să conțină doar cinematografele ce au numele „Cinema City”, au locația într-un oraș al cărui nume are o lungime mai mare decât 8, iar tipul sălilor incluse în acestea au structura ‘multi’ în nume. Filmele care rulează în aceste cinematografe vor fi afișate doar dacă rulează pe o durată mai mare de 4 luni. Aceste rezultate vor fi ordonate crescător după codul cinematorafului în care rulează.
- Join pe 4 tabele, „case”, operații pe șiruri de caractere, subcerere nesincronizată în care intervin 3 tabele, filtrare la nivel de linii, operație cu dată calendaristică, ordonare

```
select r.cod_cinema, f.cod_film, f.nume_film, d.nr_sala,  
       concat(concat(a.nume, ' '), a.prenume) as nume_actor,  
       case f.rating  
         when 6.3 then 'Nu prea apreciat'  
         when 8.0 then 'Foarte apreciat'  
         else 'Apreciat'  
       end  
       as aprecieri
```

```

from film f, ruleaza r, difuzare d, joaca j, actor a
where f.cod_film = r.cod_film and f.cod_film = d.cod_film
    and f.cod_film = j.cod_film and j.id_actor = a.id_actor
    and months_between(r.data_final, r.data_inceput) > 4
    and r.cod_cinema in (select c.cod_cinema
                           from cinematograf c, locatie l, sala s
                           where l.id_locatie = c.id_locatie
                                 and s.cod_cinema = c.cod_cinema
                                 and lower(c.nume_cinema) like lower('Cinema City')
                                 and length(l.oras) > 8
                                 and lower(s.tip) like '%multi%')

order by 1;-- 6 rezultate

```

The screenshot shows the MySQL Workbench interface with two tabs: 'Query Result' and 'Script Output'. The 'Query Result' tab contains the executed SQL code and the resulting table. The table has columns: COD_CINEMA, COD_FILM, NUME_FILM, NR_SALA, NUME_ACTOR, and APRECIERI. The data is as follows:

COD_CINEMA	COD_FILM	NUME_FILM	NR_SALA	NUME_ACTOR	APRECIERI
2	120	1010 Red Notice	25	Reynolds Ryan	Nu prea apreciat
3	220	910 The Batman	28	Pattinson Robert	Foarte apreciat
4	220	1010 Red Notice	28	Reynolds Ryan	Nu prea apreciat
5	220	1010 Red Notice	25	Reynolds Ryan	Nu prea apreciat
6	220	1710 Death on the Nile	31	Gadot Gal	Apreciat

II. Afişați numele și prenumele (concatenate) angajatului și salariul modificat pentru toți angajații care au numărul de ore egal cu numărul minim de ore din cinematograful în care sunt angajați. Rezultatele trebuie să redea angajații pentru care codul cinematografului în care lucrează este mai mare decât 120 și care au asignat un job ce conține în codul său litera 'a'. Salariul se modifică după regula următoare: dacă persoana s-a angajat în decursul anului 2017, atunci va avea o mărire de salarit cu 30%, dacă s-a angajat în decursul anului 2018, atunci va avea o mărire de salarit cu 25%, dacă s-a angajat în decursul anului 2019, atunci va avea o mărire de salarit cu 20%, dacă s-a angajat în decursul anului 2020, atunci va avea o mărire de salarit cu 15%, altfel va avea o mărire de salarit cu 10%. Rezultatele vor fi afișate descrescător, după valoarea salariului modificat.

--operații pe siruri de caractere, „decode”, operații cu date calendaristice, subcerere sincronizată în care intervin 3 tabele, filtrare la nivel de linii, ordonare

```

select concat(concat(a1.nume, ' '), a1.prenume) as nume, l1.salariu as salariu
    decode(to_char(l1.data_ang, 'YYYY'), '2017', l1.salariu * 1.3, '2018',
    l1.salariu * 1.25, '2019', l1.salariu * 1.2, '2020', l1.salariu * 1.15, l1.salariu * 1.1)
    as "Modificare salariu"
from angajat a1, lucreaza l1
where a1.cod_angajat = l1.cod_angajat
    and l1.data_demisie is null
    and months_between(sysdate, l1.data_ang) > 30
    and a1.nr_ore = (select min(nr_ore)
        from angajat a2, lucreaza l, cinematograf c
        where a1.cod_angajat = a2.cod_angajat
            and a2.cod_angajat = l.cod_angajat
            and l.cod_cinema = c.cod_cinema
            and c.cod_cinema > 120
            and lower(l.cod_job) like '%a%')
order by "Modificare salariu" desc;--6 rezultate

```

NUME	SALARIU	Modificare salariu
1 Popa Maria	6132.1	7971.73
2 Tudor Maria	5998.8	7798.44
3 Petre Cosmin	6191	7738.75
4 Nistor Viviana	6100.5	7320.6
5 Dumitran George	5789.1	7236.375
6 Banu Codrin	5772.6	7215.75

- III. Afișați pentru fiecare cinematograf ce deține mai mult de două săli sala cu indicele maxim. Rezultatele vor fi afișate descrescător după indicele săliilor.

--grupare de linii, funcții grup și filtrare la nivel de grupuri, ordonare

```

select cod_cinema, max(nr_sala)
from sala
having count(nr_sala) > 2
group by cod_cinema
order by 2 desc;--3 rezultate

```

```

select cod_cinema, max(nr_sala)
from sala
having count(nr_sala) > 2
group by cod_cinema
order by 2 desc;

```

COD_CINEMA	MAX(NR_SALA)
1	260
2	200
3	220
4	120

- IV. Afișați codul cinematografului în care lucrează fiecare angajat, codul jobului asignat, numele și prenumele (concatenate) angajatului, numărul de telefon sau lipsa acestuia și salariul angajatului. Rezultatele trebuie să respecte condiția ca pentru fiecare angajat afișat acesta să aibă salariul mai mare decât salariul minim din cinematograful în care lucrează și mai mic decât salariul mediu din cinematograful în care lucrează (bloc de instrucțiuni with).

--clauza „with”, grupare de date, funcții grup, „nvl”, operațe pe sir de caractere, ordonare, subcereri sincronizate(intervin doar 2 tabele)

```

with
sal_mediul as (select l1.cod_cinema, round(avg(l1.salariu)) as s_mediul
                 from lucreaza l1
                 group by l1.cod_cinema),
sal_min as (select l2.cod_cinema, min(l2.salariu) as s_min
                 from lucreaza l2
                 group by l2.cod_cinema)
select l.cod_cinema, l.cod_job, concat(concat(a.nume, ' '), a.prenume) as nume_angajat,
       nvl(a.telefon, 'Nu a introdus nr. telefon') as telefon, l.salariu

```

```

from angajat a, lucreaza l
where l.cod_angajat = a.cod_angajat
    and l.salariu > (select s_min
                        from sal_min
                        where cod_cinema = l.cod_cinema)
    and l.salariu < (select s_mediul
                        from sal_mediul
                        where cod_cinema = l.cod_cinema)
order by l.cod_cinema;--2 rezultate

```

The screenshot shows a SQL query builder window with the following code:

```

with
    sal_mediul as (select l1.cod_cinema, round(avg(l1.salariu)) as s_mediul
                    from lucreaza l1
                    group by l1.cod_cinema),
    sal_min as (select l2.cod_cinema, min(l2.salariu) as s_min
                    from lucreaza l2
                    group by l2.cod_cinema)
select l.cod_cinema, l.cod_job, concat(concat(a.nume, ' '), a.prenume) as nume_angajat,
       nvl(a.telefon, 'Nu a introdus nr. telefon') as telefon, l.salariu
  from angajat a, lucreaza l
 where l.cod_angajat = a.cod_angajat
       and l.salariu > (select s_min
                           from sal_min
                           where cod_cinema = l.cod_cinema)
       and l.salariu < (select s_mediul
                           from sal_mediul
                           where cod_cinema = l.cod_cinema)
order by l.cod_cinema;

```

The results are shown in a table:

COD_CINEMA	COD_JOB	NUME_ANGAJAT	TELEFON	SALARIU
1	120 CAS	Matei Andrei	0799.532.678	4672.5
2	240 ING_T	Pitco Natanael	Nu a introdus nr. telefon	3472.5

- V. Afişați numele cinematografului, complexul din care face parte (sau mesajul „este independent” în cazul în care nu aparține de un complex), orașul din care face parte, numele filmului care rulează în cinematograful respectiv, numele și prenumele (concatenate) regizorului care a îndrumat filmul respectiv și titlul acestui regizor, în funcție de numărul de premii dobândite. Titlul regizorului se obține după regula: dacă acesta nu are niciun premiu, atunci se va afișa „nepremiat”, dacă are un premiu se va afișa „puțin premiat”, dacă are 5 premii se va afișa „premiat”, iar altfel se va afișa „foarte premiat”. Rezultatele vor conține doar date despre cinematografele care s-au deschis într-o lună ce are 31 de zile și filmele care au o durată mai mare de două ore (120 minute).

--,,nvl”, „decode”, operație pe sir de caractere, join pe 6 tabele, operație cu dată calendaristică

```
select c.num_cinema, nvl(c.complex, 'este independent') as Complex, l.oras,
       f.num_film, concat(concat(re.num, ' '), re.prenume) as nume_regizor,
       decode(re.nr_premii, 0, 'nepremiat', 1, 'putin premiat', 5, 'premiat',
              'foarte premiat') as Titlu_regizor
  from cinematograf c, film f, ruleaza r, indruma i, regizor re, locatie l
 where c.cod_cinema = r.cod_cinema and r.cod_film = f.cod_film
       and f.cod_film = i.cod_film and i.id_regizor = re.id_regizor
       and c.id_locatie = l.id_locatie
       and f.durata > 120 and to_char(last_day(c.data_desch), 'DD') = 31;
```

NUME_CINEMA	COMPLEX	ORAS	NUME_FILM	NUME_REGIZOR	TITLU_REGIZOR
1 Cinema City	este independent	Constanta	The Batman	Reeves Matt	putin premiat
2 Cinema City	Park Lake	Bucuresti	Little Women	Gerwig Greta	foarte premiat
3 Cinema Arta Sibiu	este independent	Sibiu	Little Women	Gerwig Greta	foarte premiat
4 Cinema City	este independent	Baia Mare	Little Women	Gerwig Greta	foarte premiat
5 Cinema Victoria	este independent	Iasi	Star Wars: Episode III	Lucas George	foarte premiat
6 Cinema Victoria	este independent	Cluj-Napoca	Star Wars: Episode III	Lucas George	foarte premiat
7 Cinema Victoria	este independent	Iasi	It	Muschietti Andy	nepremiat
8 Cinema City	Park Lake	Bucuresti	It chapter 2	Muschietti Andy	nepremiat
9 Cinema Arta Sibiu	este independent	Sibiu	It chapter 2	Muschietti Andy	nepremiat
10 Cinema Victoria	este independent	Iasi	It chapter 2	Muschietti Andy	nepremiat
11 Cinema City	este independent	Constanta	Death on the Nile	Branagh Kenneth	foarte premiat

13. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri

I. Să se șteargă datele despre rularea tuturor filmelor ce au o durată mai mare de 2 ore și jumătate (150 minute).

--Suprimare a datelor

```
delete from ruleaza
  where cod_film in (select cod_film
                        from film
                      where durata > 150); -- 5 rows deleted
```

```
delete from ruleaza
  where cod_film in (select cod_film
                        from film
                      where durata > 150);

5 rows deleted.
```

II. Pentru toți angajații ce au codul job-ului de forma „ING...” să se modifice valoare numărului de ore lucrate, primind valoarea 8. (Să lucreze 8 ore toți inginerii).

--Actualizare a datelor

```
update angajat
  set nr_ore = 8
  where cod_angajat in (select cod_angajat
                        from lucreaza
                      where lower(cod_job) like 'ing%');--3 rows updated
```

```
update angajat
set nr_ore = 8
where cod_angajat in (select cod_angajat
from lucreaza
where lower(cod_job) like 'ing%');
```



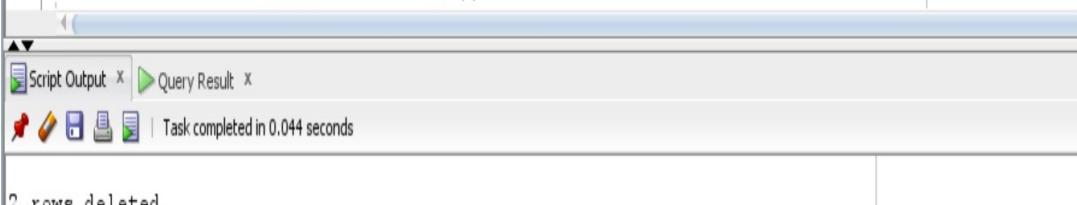
3 rows updated.

III. Să se șteargă datele despre difuzarea tuturor filmelor ce au o durată mai mare de 2 ore și jumătate (150 minute).

--Suprimare a datelor

```
delete from difuzare
where cod_film in (select cod_film
from film
where durata > 150);--2 rows deleted
```

```
delete from difuzare
where cod_film in (select cod_film
from film
where durata > 150);
```



2 rows deleted.

16. Implementarea operației de outer join și cererile ce utilizează operația division

Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outer join pe minimum 4 tabele și două cereri ce utilizează operația division.

- I. Să se afișeze codul filmului, al difuzării acestuia (un film poate avea corespondent în tabela difuzare sau nu), id-urile actorilor care joacă în acesta (un film poate avea actori atribuiți care au fost introdusi în baza de date sau nu) și id-urile regizorilor care au îndrumat acest film (un film poate avea actori atribuiți care au fost introdusi în baza de date sau nu).

--Operație de outer join pe 4 tabele

```
select f.cod_film, d.cod_difuzare, j.id_actor, i.id_regizor
from indruma i, film f, joaca j, difuzare d
where f.cod_film = i.cod_film(+)
      and f.cod_film = j.cod_film(+)
      and f.cod_film = d.cod_film(+) ;
```

The screenshot shows the Oracle SQL Developer interface. The SQL tab contains the query:

```
select f.cod_film, d.cod_difuzare, j.id_actor, i.id_regizor
from indruma i, film f, joaca j, difuzare d
where f.cod_film = i.cod_film(+)
      and f.cod_film = j.cod_film(+)
      and f.cod_film = d.cod_film(+) ;
```

The results are displayed in a grid:

	COD_FILM	COD_DIFUZARE	ID_ACTOR	ID_REGIZOR
1	10	50	10	130
2	1210	71	(null)	(null)
3	710	92	(null)	(null)
4	1510	113	(null)	(null)
5	1110	155	(null)	130
6	1110	155	(null)	210
7	1010	176	40	(null)
8	610	197	(null)	110
9	410	218	70	150
10	1310	239	(null)	(null)
11	1010	281	40	(null)
12	1210	323	(null)	(null)
13	1210	365	(null)	(null)

- II. Să se afișeze codurile filmelor care rulează în toate cinematografele ce au numele 'Cinema Victoria'.

--Implementarea operatorului „Division” utilizând de două ori instrucțiunea „NOT EXISTS”

```

select distinct cod_film
from ruleaza r1
where not exists (select *
                  from cinematograf c
                  where lower(numecinema) like 'cinema victoria'
                  and not exists (select *
                                  from ruleaza r2
                                  where r2.cod_cinema = c.cod_cinema
                                  and r1.cod_film = r2.cod_film)); --1 rezultat

```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor with the following SQL query:

```

select distinct cod_film
from ruleaza r1
where not exists (select *
                  from cinematograf c
                  where lower(numecinema) like 'cinema victoria'
                  and not exists (select *
                                  from ruleaza r2
                                  where r2.cod_cinema = c.cod_cinema
                                  and r1.cod_film = r2.cod_film));

```

In the bottom-right pane, there is a results grid titled "Query Result". It has one row with two columns: "COD_FILM" and "610". Below the grid, the status bar indicates "All Rows Fetched: 1 in 0.005 seconds".

III. Să se afișeze codul cinematografelor care rulează toate filmele care au apărut în anul 2019.

--Simularea operatorului „Division” cu ajutorul funcției „count”

insert into ruleaza

```
values(seq_cod_ruleaza.nextval, 160, 1610, '5 may 2019', '9 august 2019');
```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor with the following SQL statement:

```

insert into ruleaza
values(seq_cod_ruleaza.nextval, 160, 1610, '5 may 2019', '9 august 2019');

```

In the bottom-right pane, there is a results grid titled "Query Result". It displays the message "1 row inserted." Below the grid, the status bar indicates "Task completed in 0.035 seconds".

Am inserat în plus o valoare în tabelul „rulează” pentru a avea cel puțin o valoare întoarsă ca rezultat al acestei cereri, deoarece nu existau suficiente date care să facă posibilă corelarea datelor pentru o astfel de cerere.

```

select cod_cinema
from ruleaza
where cod_film in (select cod_film
                    from film
                    where an_aparitie = 2019)
group by cod_cinema
having count(cod_film) = (select count(*)
                           from film
                           where an_aparitie = 2019);--1 rezultat

```

```

select cod_cinema
from ruleaza
where cod_film in (select cod_film
                     from film
                     where an_aparitie = 2019)
group by cod_cinema
having count(cod_film) = (select count(*)
                           from film
                           where an_aparitie = 2019);

```

Script Output | Query Result | Query Result 1 | Query Result 2

SQL | All Rows Fetched: 1 in 0.006 seconds

COD_CINEMA
160

17. Optimizări și arbori algebrici

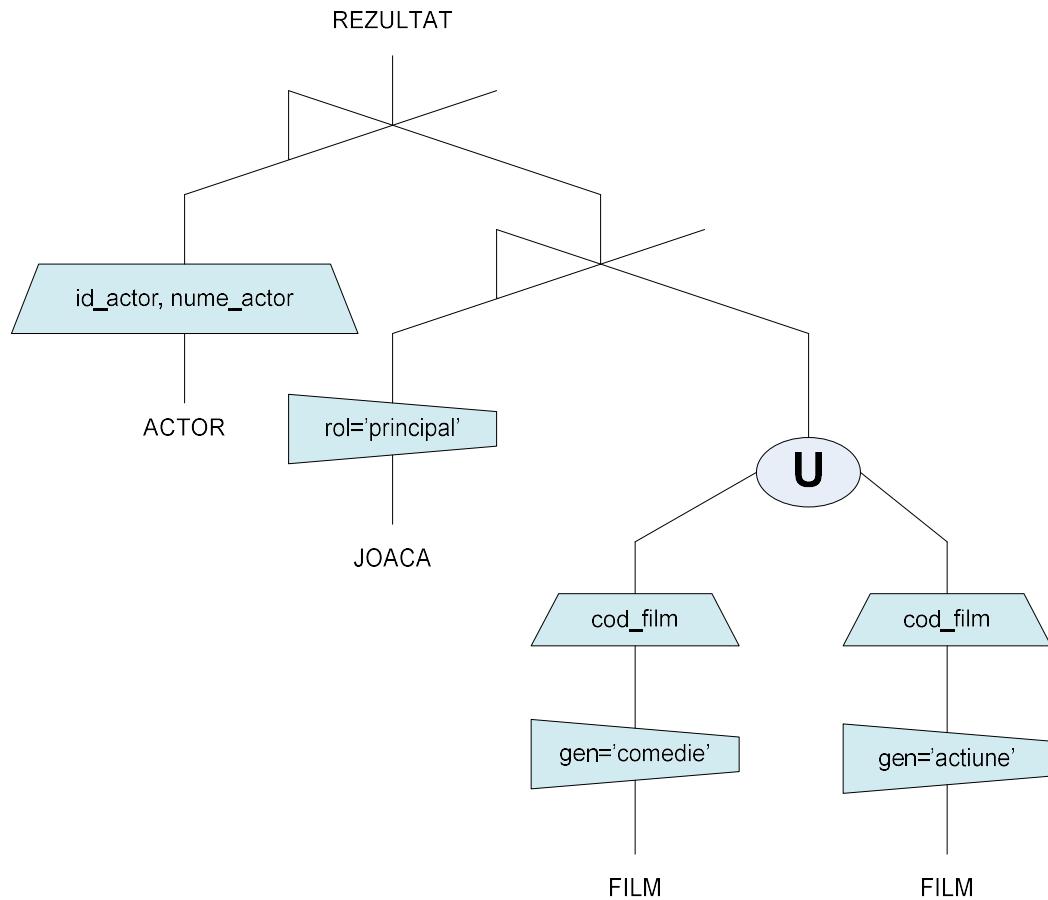
Să se afișeze id-ul și numele actorilor care au jucat în filme ce au genul „Acțiune” sau „Comedie” și au rolul ‘principal’ în acel film.

Înainte de optimizare:

```

R1 = SELECT(FILM, gen = 'actiune')
R2 = SELECT(FILM, gen= 'comedie')
R3 = UNION(R1, R2)
R4 = PROJECT(R3, cod_film)
R5= SELECT(JOACA, rol = 'principal')
R6 = SEMIJOIN(R5, R4)
R7 = SEMIJOIN(ACTOR, R6)
REZULTAT = PROJECT(R7 , id_actor, nume_actor)

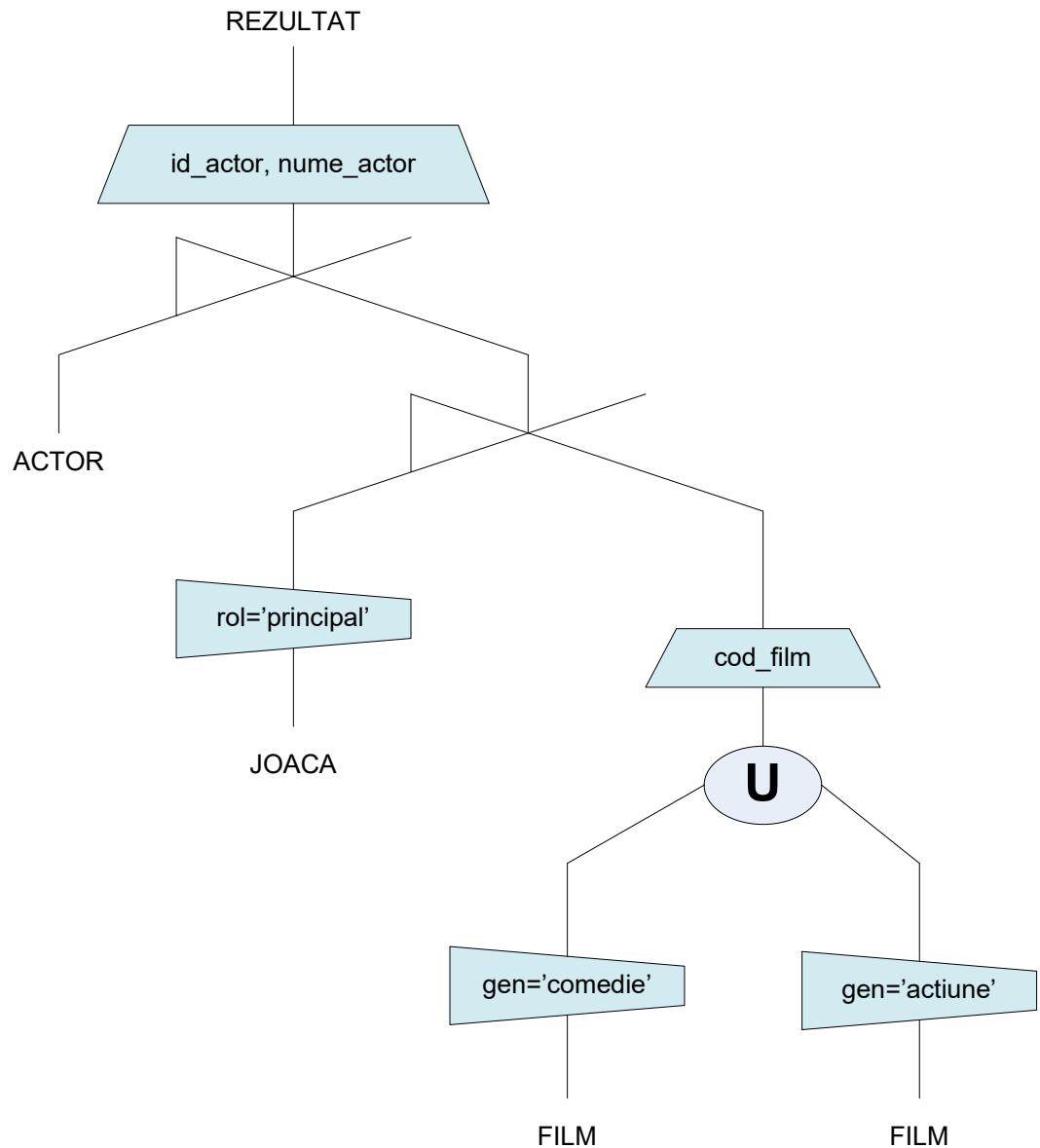
```



După optimizare:

```

REZULTAT = SEMIJOIN(PROJECT(ACTOR, id_actor, nume_actor),
SEMISEMIJOIN(SELECT(JOACA, rol = 'principal'), UNION(PROJECT(SELECT(FILM, gen =
'comedie), cod_film), PROJECT(SELECT(FILM, gen = 'actiune'), cod_film)))
  
```



18. Normalizări BCNF, FN4, FN5 + denormalizare

a. Realizarea normalizării BCNF, FN4, FN5.

I. BCFN

Se consideră relația JOACA ce leagă entitățile FILM și ACTOR. Aceasta are cardinalitatea „many to many” și va genera un tabel asociativ. Să presupunem că acest tabel ar fi avut schema relațională:

JOACA(cod_film#, id_actor#, nume_film, rol)

Pentru exemplul analizat se presupune că numele filmelor sunt unice, iar într-un film un actor poate avea un singur rol. Prin urmare, în orice moment, un film are un cod unic și un nume unic. Cheile candidat sunt {cod_film, id_actor} și {nume_film, id_actor}.

Între atributele relației există dependențele:

{cod_film, id_actor} \rightarrow {rol}

{cod_film} \rightarrow {nume_film}

Tabelul nu este în BCFN deoarece conține doi determinanți, cod_film și nume_film, care nu sunt chei candidat pentru tabelul respectiv. Ambele atrbute sunt determinanți deoarece fiecare îl determină pe celălalt.

Soluția problemei constă în divizarea relației în două proiecții conform tehnicii Casey-Delobel.

FILM(cod_film#, nume_film)

JOACA(cod_film#, id_actor#, rol)

II. FN4

Presupunem că tabela cinematograf are schema relațională:

CINEMATOGRAF(cod_cinema#, cod_angajat, cod_job)

Stim că un cinematograf poate avea mai mulți angajați și mai multe job-uri. Între atrbutele relației vor exista multidependențele:

cod_cinema# $\rightarrow\rightarrow$ cod_angajat

cod_cinema# $\rightarrow\rightarrow$ cod_job

Pentru a aduce relația CINEMATOGRAF în FN4 aceasta se va diviza în două proiecții:

CINEMATOGRAF1(cod_cinema#, cod_angajat)

CINEMATOGRAF2(cod_cinema#, cod_job)

III. FN5

Se consideră tabela LUCREAZA ce conține informații despre angajați, joburile și cinematografele în care lucrează aceștia. Se consideră schema relațională:

LUCREAZA(cod_angajat#, cod_job#, cod_cinema#)

Se presupune că fiecare angajat poate avea unul sau mai multe joburi. Fiecare cinematograf are unul sau mai multe joburi. Fiecare job se regăsește în unul sau mai multe cinematografe. Fiecare angajat poate lucra în unul sau mai multe cinematografe. Fiecare cinematograf poate avea unul sau mai mulți angajați. Fiecare job poate avea asignat unul sau mai mulți angajați.

Dacă angajatul A lucrează în cinematograful C, iar jobul J este atașat lui C, iar A lucrează pe postul J, atunci A lucrează pe postul J în cinematograful C.

Ținând cont de constrângerile impuse modelului se obțin dependențele:

{cod_angajat#, cod_job#} $\rightarrow\!\!\!-\>$ {cod_cinema}

{cod_angajat#, cod_cinema# } $\rightarrow\!\!\!-\>$ {cod_job}

{cod_job#, cod_cinema# } $\rightarrow\!\!\!-\>$ {cod_angajat}

Din cauza dependențelor formulate anterior, relația nu este în FN5. Ea se poate descompune prin proiecție în trei relații:

LUCREAZA1(cod_angajat#, cod_job#)

LUCREAZA2(cod_angajat#, cod_cinema#)

LUCREAZA3(cod_cinema#, cod_job#)

În acest caz este evident faptul că:

LUCREAZA != JOIN(LUCREAZA1, LUCREAZA2)

LUCREAZA != JOIN(LUCREAZA1, LUCREAZA3)

LUCREAZA != JOIN(LUCREAZA2, LUCREAZA 3)

Relația corectă este:

LUCREAZA = JOIN(JOIN(LUCREAZA1, LUCREAZA2), LUCREAZA3)

b. Aplicarea denormalizării, justificând necesitatea acesteia.

Fie schema relațională:

CINEMATOGRAF(cod_cinema#, nume_cinema, data_deschidere, id_locatie, oras, strada).

Relația nu este în FN3 din cauza dependenței id_locatie ->oras, strada. Se normalizează relația și se obțin schemele relaționale:

CINEMATOGRAF1(cod_cinema#, nume_cinema, data_deschidere, id_locatie)

CINEMATOGRAF2(cod_locatie#, oras, strada)

Foarte rar se va dori accesarea pașială a datelor despre locația în care se află un cinematograf, iar de fiecare dată va trebui făcut un join între cele două tabele pentru a obține datele corecte de care avem nevoie. Prin urmare structura relației din FN2 este mai simplu de folosit pentru accesarea rapidă a datelor, prin reducerea și chiar eliminarea join-urilor din cadrul interogărilor frecvente și de importanță majoră. Este mai benefic în acest caz ca în relația „copil” să se regăsească unul sau mai multe atribute din relația „părinte”, deoarece este esențial să aflăm rapid orașul și strada pe care se găsește un cinematograf.